EVOLUTION OF MEMORY IN REACTIVE ARTIFICIAL NEURAL NETWORKS

A Dissertation

by

JI RYANG CHUNG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2012

Major Subject: Computer Science

EVOLUTION OF MEMORY IN REACTIVE ARTIFICIAL NEURAL NETWORKS

A Dissertation

by

JI RYANG CHUNG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Yoonsuck Choe
Committee Members,	Frank Shipman
	Anxiao Jiang
	Takashi Yamauchi
Head of Department,	Duncan M. Walker

May 2012

Major Subject: Computer Science

ABSTRACT

Evolution of Memory in Reactive Artificial Neural Networks.

(May 2012)

Ji Ryang Chung, B.S., Seoul National University

Chair of Advisory Committee: Dr. Yoonsuck Choe

In the neuronal circuits of natural and artificial agents, memory is usually implemented with recurrent connections, since recurrence allows past agent state to affect the present, on-going behavior. Here, an interesting question arises in the context of evolution: how reactive agents could have evolved into cognitive ones with internalized memory? This study strives to find an answer to the question by simulating neuroevolution on artificial neural networks, with the hypothesis that internalization of external material interaction can be a plausible evolutionary path leading to a fully internalized memory system.

A series of computational experiments were performed to gradually verify the above hypothesis. The first experiment demonstrated the possibility that external materials can be used as memory-aids for a memoryless reactive artificial agents in a simple 1-dimensional environment. Here, the reactive artificial agents used environmental markers as memory references to be successful in the ball-catching task that requires memory.

Motivated by the result of the first experiment, an extended experiment was conducted to tackle a more complex memory problem using the same principle of external material interaction. This time, the reactive artificial agents are tasked to remember the locations of food items and the nest in a 2-dimensional environment. Such path-following behavior is a trivial foraging strategy of various lower animals such as ants and fish.

The final experiment was designed to show the evolution of internal recurrence. In this experiment, I showed the evolutionary advantage of external material interaction by comparing the results from neural network topology evolution algorithms with and without the material interaction mechanism. The result confirmed that the agents with external material interaction learned to solve the memory task faster and more accurately.

The results of the experiments provide insights on the possible evolutionary route to an internalized memory. The use of external material interaction can help reactive artificial agents to go beyond the functionality restricted by their simple network structure. Moreover, it allows much faster convergence with higher accuracy than the topological evolution of the artificial agents. These results suggest one plausible evolutionary path from reactive, through external material interaction, to recurrent structure.

DEDICATION

To my wife Jung Won for her sacrifice, love, and encouragement, and to my daughters Emily and Emma for their smiles.

ACKNOWLEDGMENTS

Foremost, I would like to express my sincere gratitude to my advisor, Dr. Yoonsuck Choe, for his mentoring and support during my doctoral journey. I am deeply grateful for all that I have learned from him, and I will keep his professional as well as personal advices as guidances for the rest of my life.

I also would like to thank my outstanding committee members, Dr. Frank Shipman, Dr. Anxiao (Andrew) Jiang, and Dr. Takashi Yamauchi for their guidance and support throughout the development of this dissertation. I would also like to thank Dr. James Caverlee for attending my oral defense on behalf of Dr. Jiang.

I would like to thank the previous and current Neural Intelligence Laboratory members, Jaerock Kwon, Huei-Fang Yang, and Timothy Mann for your valuable feedbacks and passionate discussions about my research. I would also like to thank my supportive friends Hye Jeong Kim and Sun Myung Kim for the professional and personal help.

Although I may not be able to name them all, I would like to express my deepest appreciation to all those who encouraged and supported my Ph.D. research directly or indirectly.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

A. Motivation

Artificial Intelligence (AI) was born in the summer of 1956 at the Dartmouth Conference [1]. Since then, reconstructing human-level intelligence has been its ultimate goal [2, 3, 4, 5]. However, even after a half-century of effort, reaching this goal seems difficult. There are two main reasons for this: (1) the drastically high complexity of the brain, the organ of intelligence; (2) unclear framework to understand intelligence. Before proceeding further, I feel obliged to clarify the place of my dissertation work in AI research, and to explain how I will tackle these problems to justify the methodology of my dissertation research.

1. Artificial Evolution

Until 1980's, AI showed a certain amount of success especially in various expert systems. This success was limited because designing such systems require considerable amount of human expertise and complete knowledge about input/output and the domain itself. Traditional AI tried to infer logical rules from an existing intelligent system. However, even a simple animal behavior involves intricate sensorimotor coordination and it is impossible for a designer to come up with all the rules for the kind of behavior. Moreover, because the brain is a product of continuous adaptations to

---

The journal model is *IEEE Transactions on Automatic Control.*

an ever-changing environment, the resulting structure is not guaranteed to be optimal (see e.g. [6]). This suboptimality makes it even more difficult to deduce the brain structure. Therefore, from the late 1980's, AI researchers started to think of the brain as a subsystem within a larger system of dynamic nature. In "bottom-up" approach, they advocated the importance of embodiment to show real intelligence, because sensorimotor skills are essential to higher level skills. This emergentism requires a system with basic elements that evolve to best suit its environment [7, 8, 9]. This artificial evolution approach is particularly useful in investigating the human brain and intelligence not only because the human brain is a result of biological evolution, but also because it allows us the control over the complexity.

2. Memory

AI, or intelligence in general is an extremely broad topic. It is not just about problem solving but includes various mental functions for learning, understanding, and interacting with the environment. One plausible definition of intelligence is that it is the capability of a system to achieve a set of goals in a variety of environments [10, 11]. However, such broad definition makes the attempt to build an intelligent system too audacious. Instead of such a broad definition of intelligence, this dissertation focuses on its better-defined subfunction, memory. Even though intelligence and memory are not identical, they are closely interdependent on each other [12, 13]. Memory is at the core of almost all cognitive functions, thus it has an extremely broad role. In neural networks, memory is usually implemented with recurrent feedback connections

(Figure 1). This unique structural requirement provides us with a clear distinction between *memoryless* and *memory-equipped* systems.

With the above in mind, I will attempt to show the emergence of memory in artificial neural networks through artificial evolution, the AI version of the emergence of a key component of intelligence in animal brain throughout natural evolution. I hope this effort can contribute to our understanding of human intelligence.

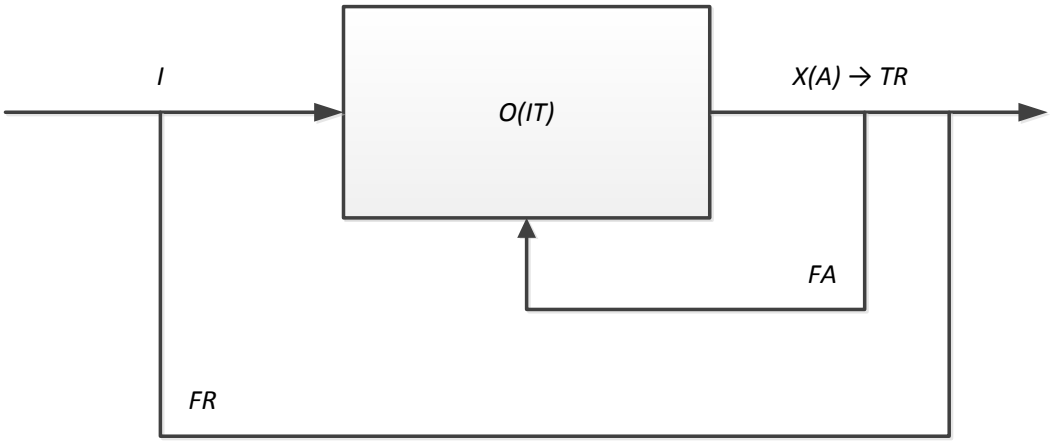


Fig. 1. **Piaget and Inhelder's Memory Model.** I = input(entry); O = organization (system of internal transformations); $X(A)$ = output or actions leading to TR = transformations of reality; FR = feedbacks based on the results of TR; FA = feedbacks based on $X(A)$. Redrawn from [13].

B. Main Research Question

In the neuronal networks of natural or artificial agents, memory is usually implemented with recurrent connections. It is well established that for a neural network to show some level of memory, it needs recurrent connections [14, 15]. Also, the brain does not have a strictly hierarchical organization: it has rich sets of reciprocal

projections and loops (see e.g. [16]).

However, the nervous system of primitive creatures may have been limited to a feedforward topology, thus exhibiting only reactive (reflexive) behavior [17, 18]. How could this kind of primitive nervous system evolve to be equipped with a full memory system, conferring the agent the ability to perform cognitive tasks? A clue to this question can be found in the olfactory system. The most immediate sensory information that led to internalized memory may have been olfaction, or chemoreception. It is one of the most fundamental capacities already existing in the simplest early animals [19]. Also, even the earliest creatures must have had chemical dropping behavior in the form of secretion or excretion because these are natural bi-product of metabolism [20]. Combined, the two functions could have led to a memory capacity. Wadhams and Armitage showed that a highly specialized chemotaxis sytem exists in primitive creatures like the bacteria, which can distinguish more than 50 different types of proteins and a large portion of their genome is dedicated to the encoding in the chemotaxis system [21]. Furthermore, Tang-Martinez proposed the possibility that self-generated olfactory cues could have been used to develop elementary intelligence to recognize kin by *phenotype matching* [22]. Because this discrimination of kinship is essential for the survival of the organism, even sea squirts without brain can identify their kins using similar chemical cues [23]. Kin recognition does not require any long term memory: simple genetic encoding will suffice (e.g., the MHC-based recognition in mice and tadpoles). It is also suggested that chemical cues may have evolved into communication signals [24, 25]. Sorensen and Stacey(1999) and Wy-

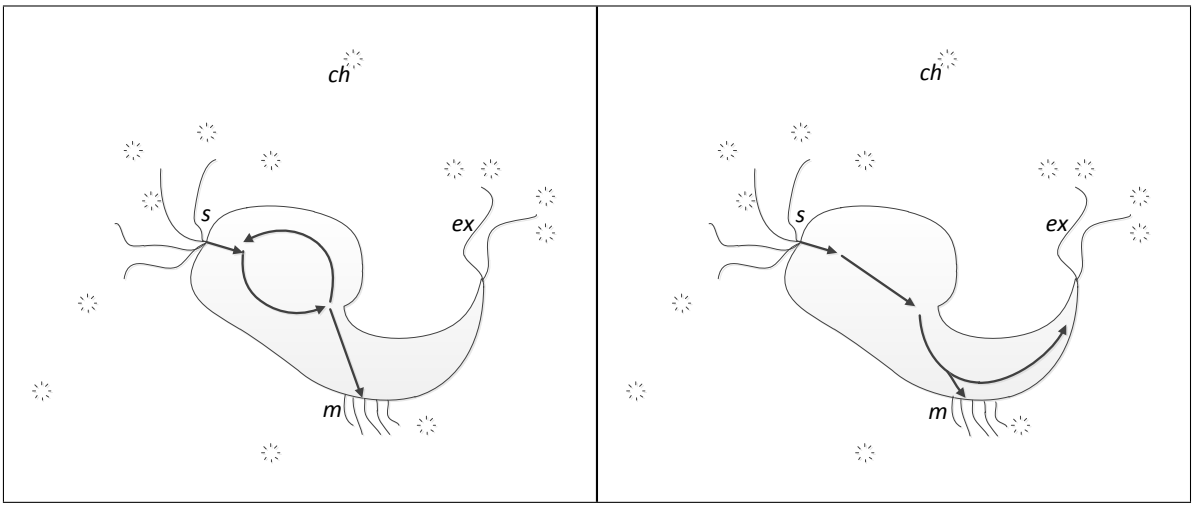
att(2003) contemplated that leaking hormones or other metabolites could have been the origin of pheromone. Furthermore, cognitive uses of such chemical signal could have evolved into internalized neuromodulators (cf. [26]). These studies support the biological feasibility of the two functions discussed above; olfaction and secretion.

C. Approach

To investigate the possibility of olfaction as the evolutionary origin of memory in artificial neural networks, I adopted two memory tasks: (1) ball catching task (Chapters III and V); (2) food foraging task (Chapter IV). In the tasks, I will compare the performances of two types of agents: (1) recurrent neural network; (2) feedforward neural network with olfactory signaling/sensing capability (*dropper network*). *Dropper network* and recurrent network represent the brain before and after the evolution of memory function (Figure 2). Comparable performance between these two types of agents will attest that memory can be formed via external (chemical) material interaction, without internalized recurrent connections. Moreover, demonstrating higher evolutionary benefit of the dropper agents will further fortify the possibility that the external material interaction can indeed be the evolutionary origin of memory.

D. Outline

This dissertation will be organized as follows. Chapter II will introduce the background of my research. Then, two artificial evolution simulations on memory tasks with increasing difficulties will follow (Chapters III and IV). These chapters will show



A. Recurrent Animal B. Dropper Animal

Fig. 2. **Conceptual Basal Animals Implementing Recurrent Neural Network and Dropper Network.** A. Animal with recurrent neural network. B. Animal with *dropper network*. *ch*. chemicals in the environment. *s*. sensory system. *m* motor system. *ex* excretory system. The key differences between the two animals are in their neural structure and the motor control of the excretory system (see Chapter II).

the emergence of memory-like behavior in the memoryless *dropper network*. The next chapter (Chapter V) will contain an artificial evolution simulation demonstrating the evolutionary advantage of the *dropper network* over the recurrent one. Finally, Chapter VI concludes the whole dissertation with general discussions and future directions.

CHAPTER II

BACKGROUND AND RELATED WORK

A. Olfaction and Memory

In this chapter, I will introduce existing research from related disciplines supporting the relationship between olfaction and memory.

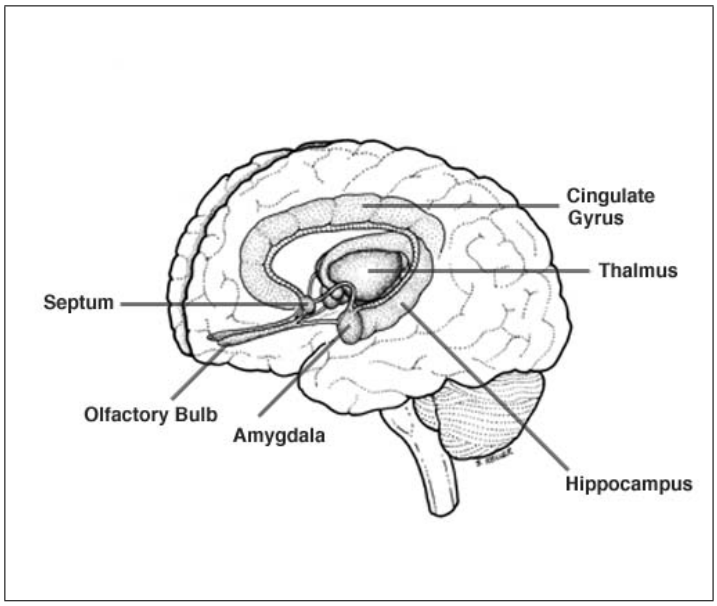
1. Evolutionary Relationship between Olfaction and Memory

Olfaction is known to be the earliest sense because of the primitiveness of the sensor and the universality in the animal phyla [19, 27]. Evolution brought animals from water to land [28]. Coupled with the evolution of the animal brain, and hence their intelligence, the change of environment in evolution suggests something important. Unlike ground-based animals rely mostly on volatile chemicals transmitted via air, water is a better medium for chemical communication as it delivers much heavier odor cues. This environmental change forced more evolved animals to less rely on the olfaction [29]. Niimura and Nei (2005) showed that the diversity of olfactory receptor genes has been reduced as animals evolve from fish to amphibians, and from amphibians to mammals and avians. Such deterioration of olfactory system continued in mammalian evolution [30]. Nature selects based on adaptability, which affects survival. If olfactory sense is not beneficial, it is meant to die out. However, olfaction has a unique characteristic that can greatly benefit one's survival. Unlike other sensory signals, odor cues can last and "transmit into the future" [31]. This

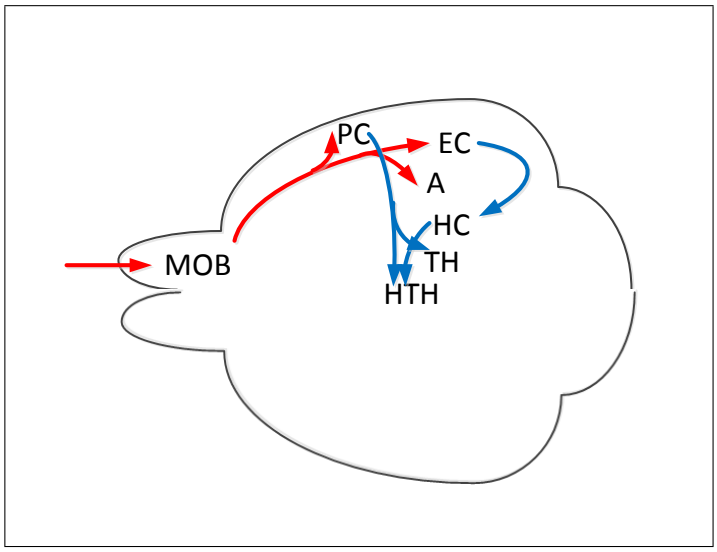
temporal aspect allows animals to access past odor signals, which is the very function of memory. Researchers including Moulton and Giland et al. noticed the relationship between the reduction of olfactory system and positive evolution in other brain areas [32, 33]. Because both olfaction and memory share temporal function, I believe that the deterioration in olfactory system has been compensated by memory system throughout evolution. Indeed, there exist a host of findings from various research domains suggesting close relationship between olfaction and memory.

2. Neurophysiological Relationship between Olfaction and Memory

An interesting connection between olfaction and memory can be found in the relationship between the olfactory system and the hippocampus in the mammalian brain. Researchers including [27, 34] have shown the anatomical and functional proximities between the olfactory bulb and the hippocampus. The olfactory bulb is a structure in mammalian forebrain where the olfactory pathway starts, and the hippocampus is a brain region involved in spatial memory. Anatomically, the olfactory bulb is located only a couple of synapses away from the hippocampus. Moreover, unlike other senses, olfactory sense bypasses the thalamus and directly feeds into the limbic system which includes the hippocampus (Figure 3). Because of these facts, hippocampus was once believed to be part of the olfactory system. Moreover, unlike any other sensory neurons, olfactory neurons are unmyelinated, thus making olfaction the slowest sense [27]. This provides olfactory sense a temporal aspect, like memory does, because it persist the longest in the brain.



A.Limbic System



B. Major Olfactory Signal Pathway

Fig. 3. **Olfactory System.** A. Limbic system is shown. Olfactory bulb is located proximal to the Amygdala and Hippocampus. Adapted from [35]. B. Major olfactory signal pathway. Red arrows indicate the main olfactory pathway and blue arrows intracortical projections. Abbreviations: MOB, main olfactory bulb; PC, piriform cortex; A, amygdala; EC, entorhinal cortex; HC, hippocampus; TH, thalamus; HTH, hypothalamus.

3. Functional Relationship between Olfaction and Memory

This unique adjacency between the olfactory and the memory system is further emphasized in Proustian retrieval of autobiographical memory [36, 37, 38]. This is a phenomenon that autobiographical memory can be best retrieved by olfactory cues. Researchers believe that the possible involvement of the olfactory bulb in memory consolidation, due to its anatomical closeness to the amygdala and the hippocampus, makes olfaction also involved in memory formation [39]. Moreover, researchers have shown that the memory system related to olfaction is separate from the memory system for other sensory modalities [40, 27, 41].

4. Embryological Relationship between Olfaction and Memory

In addition, embryological evidence suggests a genetic kinship between the olfactory bulb and the hippocampus. In an effort to show that neurogenesis is not restricted to the embryonic period but also occurs in the adult mammalian nervous system, Altman *et al.* and Frisén *et al.* found that neurogenesis in adults is most often observed in the olfactory bulb and the hippocampus, but rarely elsewhere [42, 43]. Machold *et al.* and Palma *et al.* tested the requirement for hedgehog signaling in the telencephalon and the subventricular zone [44, 45]. They mutated the Sonic hedgehog (Shh) gene, a ligand in the hedgehog signaling pathway regulating vertebrate organogenesis, to examine the resulting postnatal abnormalities. They found that Shh mutation affected the hippocampus but surprisingly it also affected the olfactory bulb. These results altogether imply a close genetic relationship between the olfactory bulb and

the hippocampus.

B. Related Works

This dissertation is related to a wide range of literatures such as robotics, machine learning, and artificial life. This section presents an overview of the work in these related fields. Reviewing these fields also identifies the limitations in the current approaches.

1. Robotics

Two streams of biologically inspired robotics research are related: behavior-based robotics and evolutionary robotics. Here, I provide a brief summary and evaluation.

a. Behavior-based Robotics

Mataric provides a good overview of the concept of behavior-based robotics [46]. Behavior-based robotics is a branch of robotics with reactive controllers consisting of a set of simple behavioral primitives. The states represented in behaviors enable the behavior-based system to generate practical artificial behaviors. Instead of nonintrinsic computations used in conventional robotics, this behavior-based methodology exploits the principle of *stigmergy* in a sense that the current behavior is affected by previous ones. The simplicity makes behavior-based robots better candidates for computationaly expensive tasks in dynamic environments [47]. However, this method requires designer's intuition to define the behavioral architecture. This approach is biologically implausible, since the designer's intuition cannot be always correct and it

can block the chance for agents to develop unexpected yet effective behavior(cf. [48] on the emergence of unexpected strategy.) This might be the reason why behavior-based systems have difficulty scaling up to handle complex problems. Also, they lack the explanation on how the behavioral architecture is learned or developed.

b. Evolutionary Robotics

Unlike behavior-based robotics that requires designer's intrusion to set up basic behaviors, desired behaviors are acquired in a self-organizing manner in evolutionary robotics [49]. Mimicking biological evolution, it follows the Darwinian principle of selective reproduction of the fittest. Randomly created initial population with different genotypes of control system (usually neural networks) are evaluated by fitness score and those with higher scores are selected to produce offsprings via crossover and mutation. This process is repeated until desired progress in the fitness is reached. This methodology requires less human intervention and results in the evolution of adaptive agents with emergent behaviors/strategies [50]. This method is commonly used for evolving memory-like behavior in feed-forward, thus reactive agents. However, learning is constrained by the preset phenotype in conventional simulated evolution and this is not enough to show how intelligent agents are evolved to equip feed-back connections, the internalized memory.

2. Topological Evolution

To cope with the structural constraints of neural network, methods evolving both neural network topologies and weights have been developed [51, 52, 53, 54]. Evolv-

ing neural network topologies can overcome the functionality limitation and result in significant performance gain. Amongst many, Stanley and Miikkulainen argued that their NeuroEvolution of Augmenting Topologies (NEAT) can solve the problems inherent to topological evolution while maintaining minimal dimensionality [54].

Complex tasks are not easy to solve using typical neuroevolution because the search space can be too large and defining fitness functions becomes too difficult. An effective strategy for such case is to decompose the global task into multiple subtasks and to let the agents to learn each subtask one after another [55, 56]. However because the network size in NEAT or other topological evolution methods keeps increasing, the final network structure after learning all the subtasks can be unnecessarily large with superfluous connections [57]. Moreover, because each subtask can generate distinct network topology [58], different sequences of subtasks will yield totally different network topology at the end, even though their functions are the same.

3. Pheromone Agents

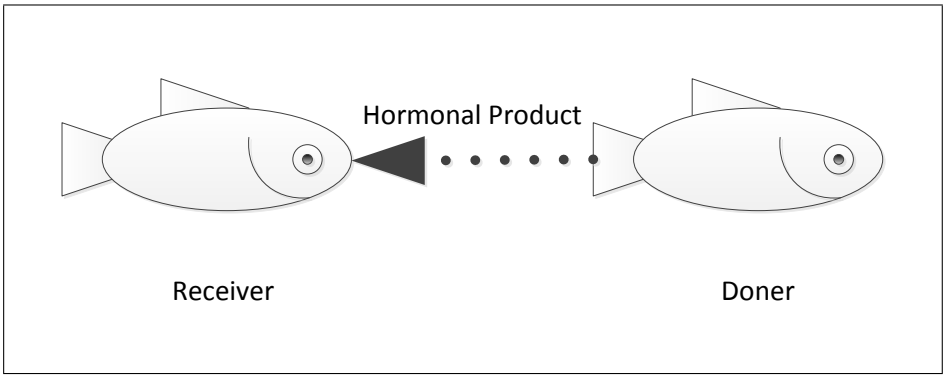
Using pheromone as a communication medium is an effective strategy proven by nature. From the fact that pheromones are used more by insects than by higher animals, we can infer that it can be used to increase the capacity of the simple brain. Inspired by this, numerous studies have been performed with swarms of simple agents using virtual pheromone [59, 60, 61, 62, 63, 64]. However, they fail to address a substantial property of the pheromone. Pheromones have two important properties: spatial and temporal. The location and time of a pheromone drop denote the locus and instant of

importance respectively. Nontheless, most studies on pheromone agents overlook the temporal property by making the agents throw pheromone unconditionally. Moreover, their investigations are limited to social learning. Pheromone can indeed serve as a medium of communication between or among multiple agents. However, a perspective lacking in the existing studies is that it also enables an agent to communicate with oneself in different points of time, just like in memory.

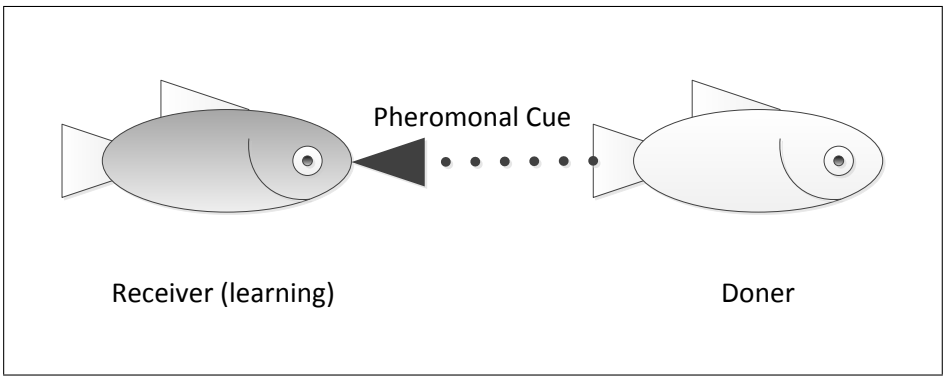
Sorensen and Stacey (1999) suggested an interesting model for the evolution of pheromone communication [24] (Figure 4). In the model, it requires two optimization stages, one for the receiver and the other for the doner of the pheromone. The mutual benefit from communication requires co-evolution of both doner and receiver. As memory can be thought of as a communication with oneself in different points of time, the self-communication of the dropper network with dropper/detector mechanisms can be seen as almost a single party version of this model (Figure 5).

C. Summary

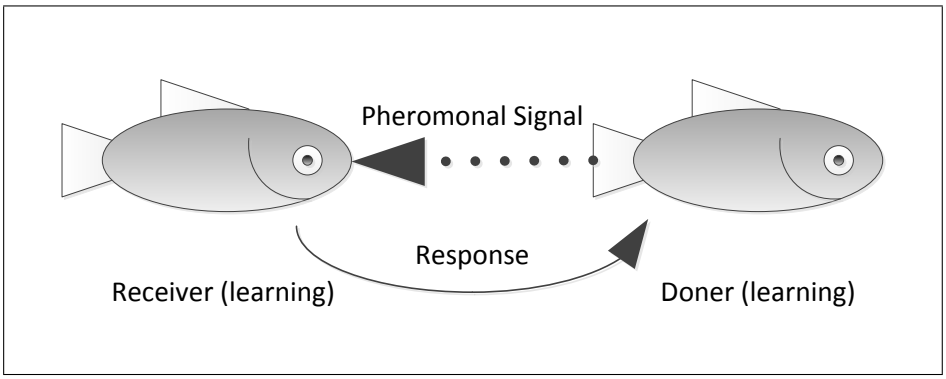
In this chapter, I introduced existing research motivating this dissertation. Also introduced are the researches that share common grounds with the current work. These evidence, relevance, and methodologies necessitate the study on the evolution of memory. In the following chapters, I will introduce a series of experiments that possibly enlighten the evolutionary origin of memory.



A. Preadaptation

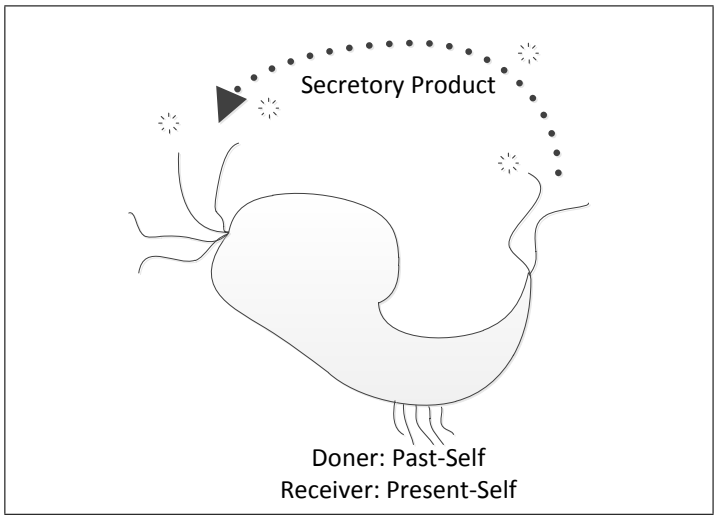


B. Spying

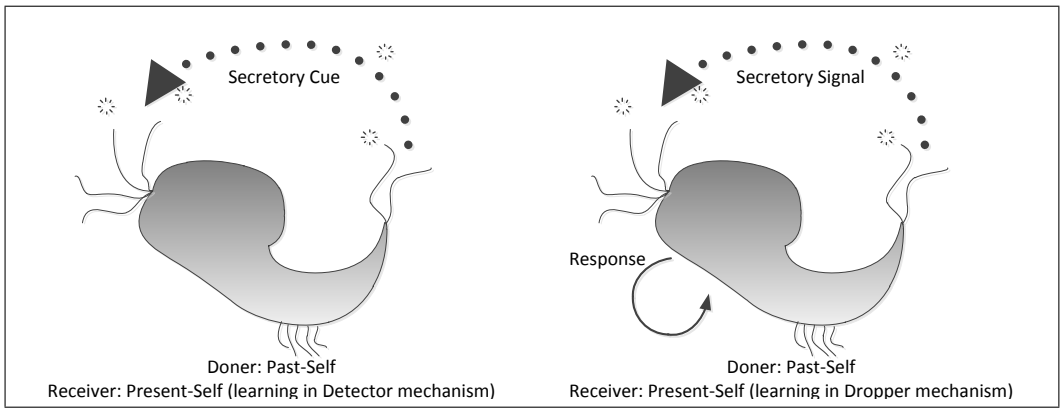


C. Communication

Fig. 4. **Evolutionary Model for Pheromone Communication.** A. Hormonal product released from the doner, but there is no mechanisms for detection or response in the receiver yet. B. Receiver's olfactory function evolves to detect/respond to the pheromonal cue from doner. Receiver benefits and release of hormonal product is not changed yet. C. Doner's pheromone production evolved to optimize its benefit from the receiver's response. Adapted and from [24].



A. Preadaptation



B. Spying & Communication

Fig. 5. **Evolutionary Model of Memory (Self-communication).** A. Preadaptation: Secretory product released. B. Spying (left) and Communication (right) concurrently evolves.

CHAPTER III

TASK I: CATCHING FALLING BALLS IN 1D

A. Task Description

In order to test if the use of external markers can work as well as the use of recurrent memory, I used a delayed-response task inspired by [65, 66] (Figure 6). An agent controlled by a neural network moves horizontally at the bottom of the 2D environment while trying to catch falling balls (the movement is in 1D). The environment size was 400×400. The agent had an array of five range sensors with limited radius (=200). Two balls are dropped from the top at different speeds. They can be sensed if they come into contact with the range sensors. The goal of the agent is to catch both balls. The initial position of the balls can vary within the range of the agent's sensors, with two constraints: first, they are to be located on the two different sides (left and right) of the agent's initial position and they must also be horizontally separated far enough to meet the memory requirement of the task (if the balls are too close, they will remain within the sensor range at all times). Memory is necessary to be successful in this task, as can be seen in Figure 6.

B. Methods

Three agents, each controlled by a different type of neural network, were used in the experiment: (1) feedforward network, (2) recurrent network, and (3) feedforward network with external marker dropper/detector ("dropper network").

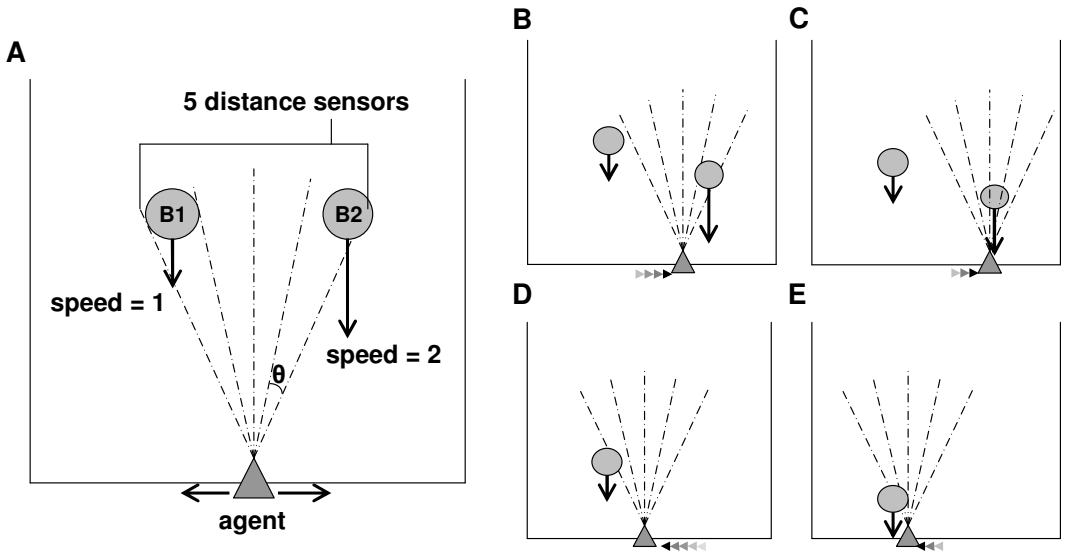


Fig. 6. **Ball Catching Task.** The task illustrates a scenario where memory may be needed. A. The agent is initially placed at the horizontal center of the environment. Two balls are falling at different speeds (the right ball is faster in this example). B. When the agent moves to catch the faster ball, the slower one goes out of the agent's view. C. The agent catches the first ball. D. The agent must move back to the slower ball without any input from the range sensors (this will require memory of where the other ball was when the agent lost contact of it). E. The agent catches the second ball.

1. Feedforward Network

The first controller is made of a simple fully connected feedforward neural network containing three layers - input, hidden, and output (Figure 7). Five inputs of the input layer were from five sensors. Sensor inputs were inversely proportional to the distance between the agent and the ball detected by the sensor:

$$I_i = 1 - Dist_i/Length_i \tag{3.1}$$

where $Length_i$ denotes the length of the range sensors which was set to 200 for all 5 sensors and $Dist_i$ is the distance between the agent and the ball detected by the i-th sensor. $Dist_i$ was 200 when no ball is detected, which makes $I_i = 0$. The input values are propagated to the hidden layer and then to the output layer. (Note that bias units were not used [in all three agents] to avoid default behavior that can artificially crank up the performance.) Activation values of the two outputs (O_1 and O_2 in Figure 7) decide the movement of the agent (the horizontal location in the next time step, Loc). The agent takes a unit-sized step to its left if $O_1 > O_2$, to the right if $O_1 < O_2$, and stays if $O_1 = O_2$. The speed of the agent is fixed to a unit distance per time step. The equations for this baseline agent take the following standard form (see e.g., [67]):

$$\begin{aligned} H_j &= \sigma\left(\sum_{i=1}^{N_{\text{in}}} v_{ji} I_i\right) \qquad j = 1, ..., N_{\text{hid}} \\ O_k &= \sigma\left(\sum_{j=1}^{N_{\text{hid}}} w_{kj} H_j\right) \qquad k = 1, ..., N_{\text{out}} \\ Loc(t+1) &= \begin{cases} Loc(t) - 1 & O_1(t) > O_2(t) \\ Loc(t) + 1 & O_1(t) < O_2(t) \\ Loc(t) & O_1(t) = O_2(t) \end{cases} \end{aligned} \tag{3.2}$$

where I_i, H_j and O_k are the activations of the i-th input, j-th hidden, and k-th output neurons; v_{ji} the input-to-hidden weights and w_{kj} the hidden-to-output weights; $\sigma(\cdot)$ the sigmoid activation function; and N_{in}, N_{hid}, and N_{out} are the number of input, hidden, and output neurons whose values were 5, 3, and 2 respectively. The unit-distance of the movement of the agent was set to 1.

This agent is expected to fail in the given task because it cannot remember the

existence of the slower ball at the time it catches the faster one. Right after catching the faster ball, the agent receives no input to drive further movement (Figure 6C). The purpose of showing this agent is just to provide the basic structure for the other agents and establish baseline performance.

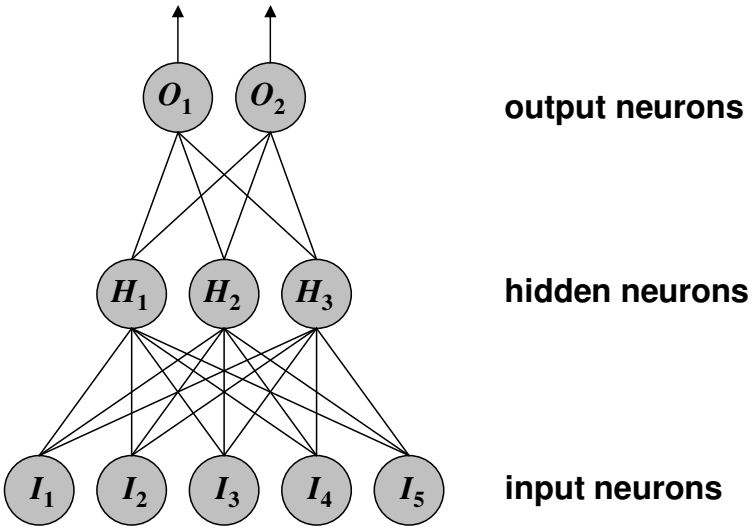


Fig. 7. **Baseline Feedforward Agent.**

2. Recurrent Network

A memory mechanism is built into the second agent by recurrent connections feeding previous hidden states back to the hidden layer (Figure 8). This type of recurrent neural network known as Elman Tower is an extension of the basic Elman network, which is one of the most commonly referenced recurrent networks [15]. Also, other researchers used recurrent networks to model the hippocampus (see e.g. [68]). Because it is one of the simplest recurrent networks, the Elman network is a good candidate for an immediate descendent of feedforward networks, as a possible first step in evolution

from a memoryless to a memory-capable system. Usually the more the number of hidden state feedbacks from the past, the more powerful the network is. The number of hidden state feedbacks was either 3 or 7, depending on the experiment. Recurrent connections can be defined as in the equation below:

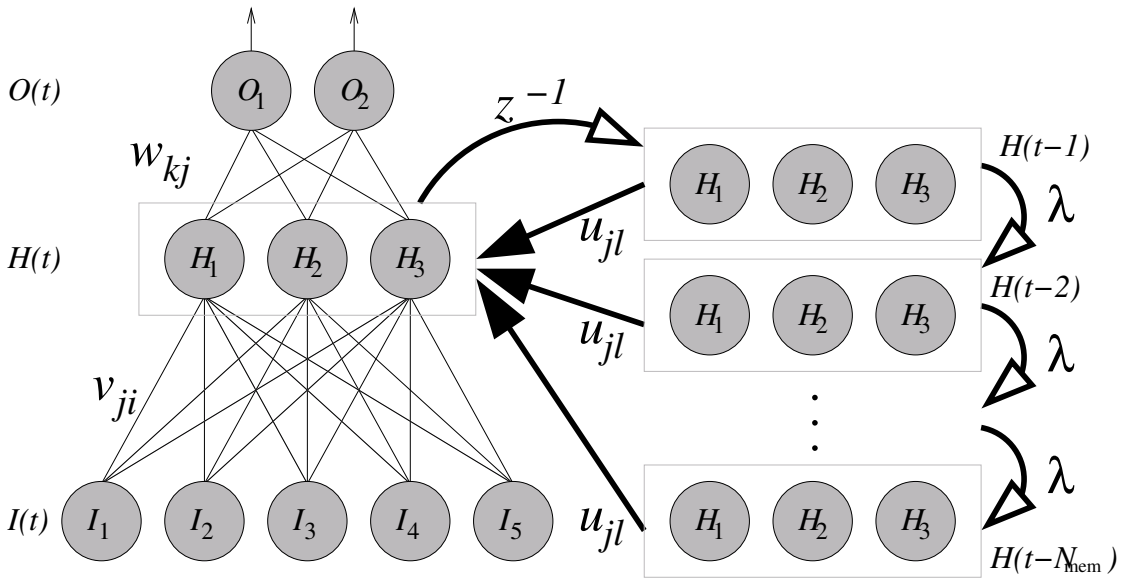


Fig. 8. **Memory-equipped Recurrent Agent.** N_{mem} previous hidden state vectors are fed back to the hidden layer (filled arrows). The open arrows indicate a 1-step delayed copying operation (λ indicates the decay rate). Hidden-to-output connection is the same as that of the baseline agent. See the text for the definition of other terms.

$$H_j(t) = \sigma\left(\sum_{i=1}^{N_{\text{in}}} v_{ji} I_i(t) + \sum_{m=1}^{N_{\text{mem}}} \sum_{l=1}^{N_{\text{hid}}} \lambda^m u_{jl}(m) H_l(t-m)\right) \qquad (3.3)$$

$$j = 1, ..., N_{\text{hid}}$$

where $u_{jl}(m)$ is the recurrent connection weight from the m-th previous hidden state vector $H_l(t-m)$. A constant decay rate ($\lambda = 0.7$) was used to penalize the effect of older state vectors, that is, $H(t-m) = \lambda H(t-(m-1))$. All other terms were identical to those explained above.

3. Dropper Network (Feedforward Network with Dropper/Detector)

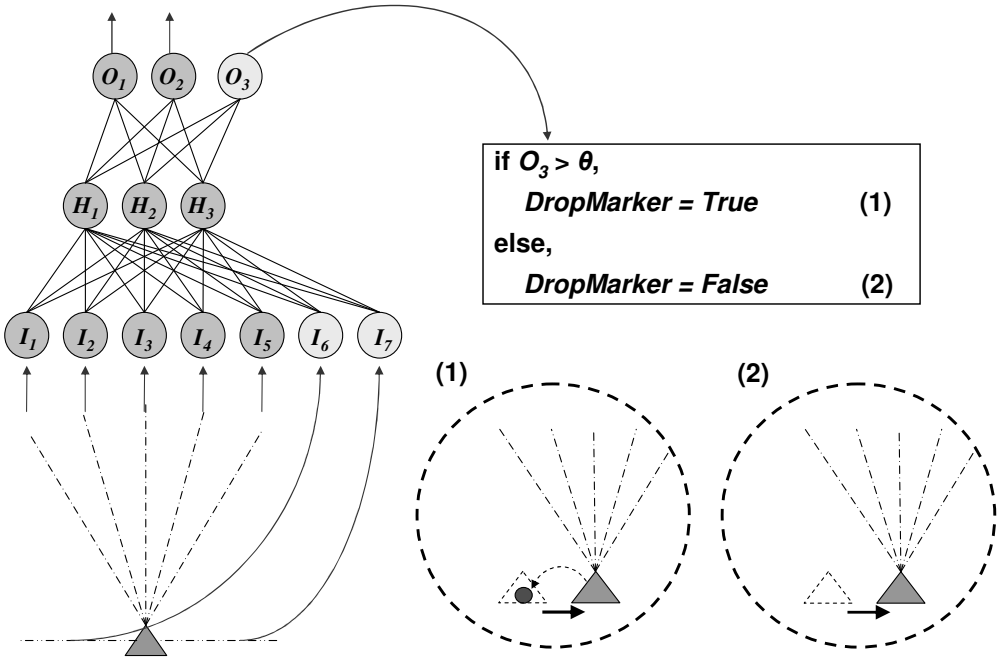


Fig. 9. **Feedforward Agent with External Marker Dropper/Detector ("Dropper Network").** The architecture is identical to the feedforward agent, with the difference being the two additional inputs to detect the external markers (I_6:left, I_7:right) , and an added output for dropping markers (O_3).

Finally, the last one, the one of my interest, is the agent using external markers (imagine leaving a bread-crumb trail). The underlying neural network is a feedfor-

ward network, identical to the baseline agent. The only difference is that it has two more inputs and it has one more output (Figure 9). The additional inputs are from supplementary sensors which can detect external markers (I_6 [marker detected on the left] and I_7 [marker detected on the right] in Figure 9). The additional output (O_3) is for the motor action to drop a marker in the environment. Because the primitive animal this agent is modeling already has external sensors, there is no evolutionary overhead to have additional mechanisms like these. That is, primitive animals may use already existing sensors to sense external markers in the environment. Throwing markers is a mere modeling of a behavior they may already have, e.g., excretion. Below is how this additional output works:

$$DropMarker = \begin{cases} \text{True} & \text{if } O_3 > \theta \\ \text{False} & \text{otherwise} \end{cases} \tag{3.4}$$

If the value of O_3 is greater than the threshold θ, the agent drops a marker in its current position before it moves to the next position. This threshold is not fixed but is also learned through genetic search. Note that this whole scheme could be seen as just another round-about way of adding recurrence, however it is a simulation of an *indirect* external loop and thus is different from direct, internal, recurrent neural circuits. The recurrence created by the dropper network is *indirect* because it does not use the dynamics of the network, but uses sensory-motor interaction, and is thus *second-hand*. Moreover, while it is true that I_6 or I_7 activates as O_3 goes on, O_3's on/off condition is relative to an evolved threshold value, and the dropping event also specifies the location of the marker in the environment (this latter aspect is more

prominent in the foraging task). Therefore, this is *indirect*, different from the *direct* neuronal relay in a recurrent network architecture.

C. Experiments and Results

The learning of connection weights of the agents is achieved through genetic search (neuroevolution). The fitness for an agent is set inversely proportional to the sum of the horizontal separations between itself and each ball when the balls hit the ground. The tasks were given to each agent 12 times (24 balls), where the ball to the right of the agent is falling faster in the first 6 tasks and vice versa for the 6 latter tasks. Fitness values of the 12 tasks are added to form the overall fitness for the agent. Best performing agents of the population in the current generation survive to the next generation. One-point crossover with probability 0.9 and mutation with rate 0.04 was applied to these best-performing agents to modify the connection weights of the current population. The halting criterion of the evolution was when any single agent in the current population is successful in catching more than 23 balls out of 24 (about 96% success rate). When the evolution step reaches a preset maximum, a fresh new agent pool was created by randomizing their weights and a new evolution process initiated.

In addition to the initial comparison of the performance between the three agent types, I wanted to know the effect of the capacity of the external marker detectors on the performance of the external-marker-using agent. Therefore, two external-marker-using agents with different marker detector range (one with the same length as the

distance sensors, the other with 1/4 the length) have been tested. For fair comparison, I also varied the capacity of the memory-equipped agent (one with the memory order N_{mem} [the number of hidden state vector feedbacks] set to 3, and the other to 7).

As expected, the baseline agent in Figure 7 could never succeed in the task. It could catch only one of the balls (i.e., success rate was 50% at best). The agent stops upon catching the first ball because it does not receive any further input at that point (Figure 6C). Because this agent was tested only to emphasize the memory requirement of this delayed-memory task, I will not show the detailed result here. The following results are from 4 groups of agents (2 network types $\times$ 2 capacity differences). Table I summarizes the 4 groups.

Table I. Ball-catching Task Experiment Description

Agent	Mechanism	Capacity/Range
SM (Short Memory)	Recurrent Network	Low
LM (Long Memory)	Recurrent Network	High
SS (Short Sensor)	Dropper Network	Low
LS (Long Sensor)	Dropper Network	High

Figures 10 and 11 show the agents' trajectories in the four experiments. In the figures, the y-axis represents time and the x-axis relative horizontal location (0 marks the initial position of the agent, and - and + the right and the left of that position). The balls on the left falls faster in the first 6 tests and the balls on the right falls faster in the 6 remaining tests. All the agents in the 4 different groups were successful in catching all 12 pairs of balls.

I compared the performance by running 200 pairs of the balls (100 in each faster-left and faster-right ball cases) for 5 agents in each experiment. Results show that all

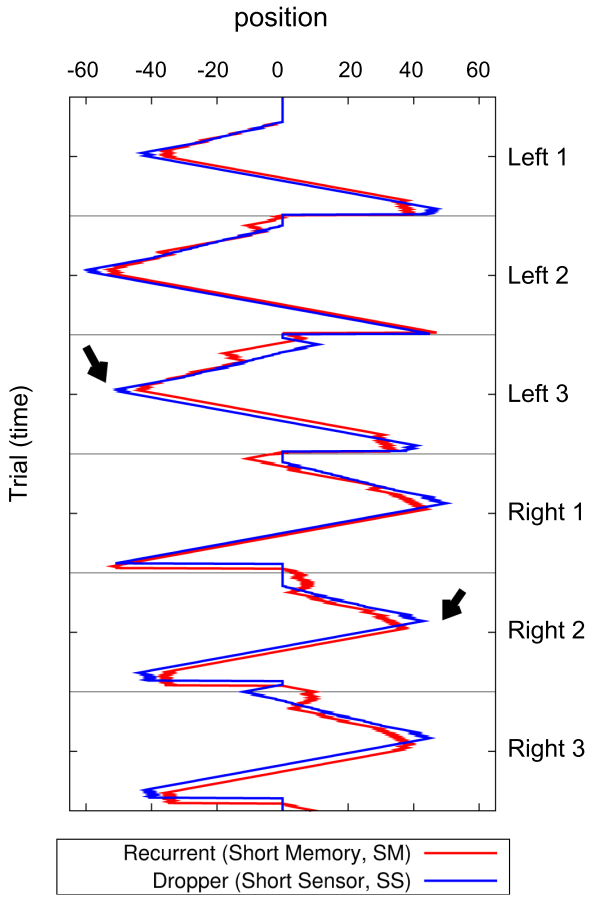


Fig. 10. **Trajectories of the Agents with Short Capacity/Range.** The trajectories of the recurrent agents with low memory capacity (red curve) and the dropper agents with short marker sensor range (blue curve) are shown. Six trials are shown, with three "fast left ball" (marked Left 1 to 3) and three "fast right ball" conditions (marked Right 1 to 3) for 200 time steps. Agents using external markers (SS) always slightly overshoot (few examples marked by $\rightarrow$) the horizontal position of the first ball they catch (blue curves).

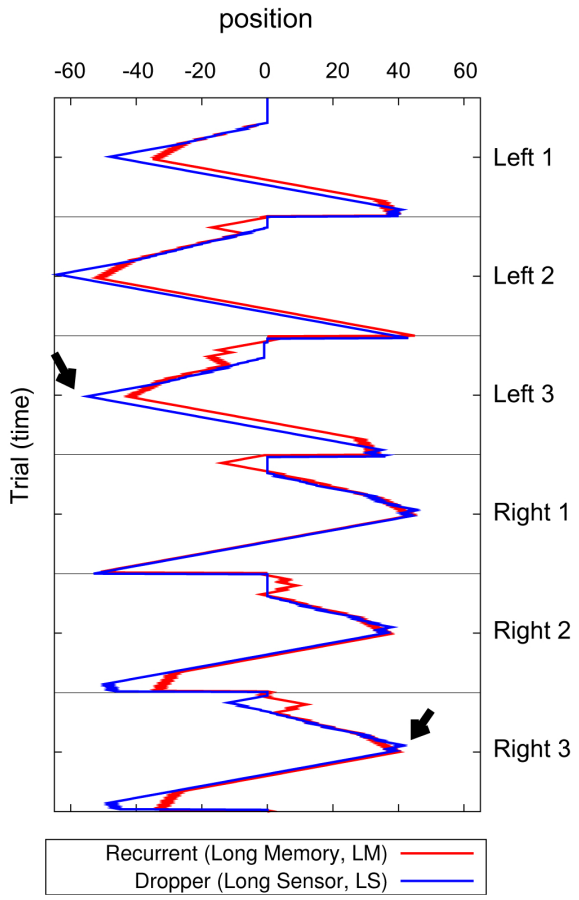
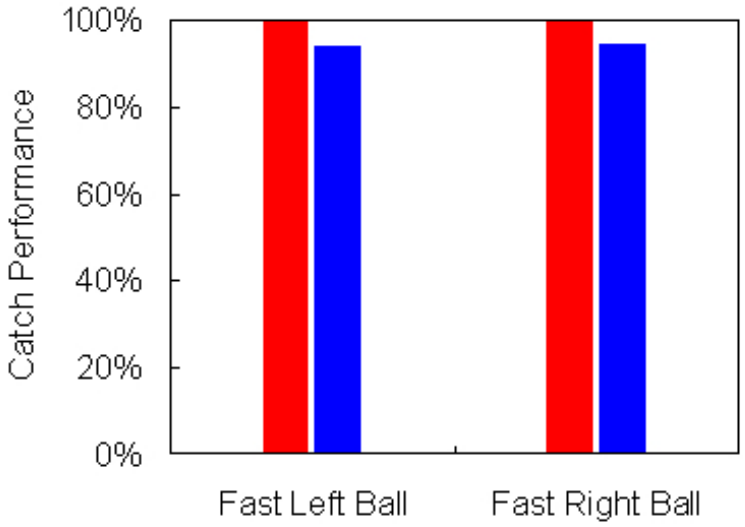


Fig. 11. **Trajectories of the Agents with Long Capacity/Range.** The trajectories of the recurrent agents with high memory capacity (red curve) and the dropper agents with long marker sensor range (blue curve) are shown. Again, agents using external markers (LS) always slightly overshoot (few examples marked by $\rightarrow$) the horizontal position of the first ball they catch (blue curves). See Figure 10 for plotting convention.

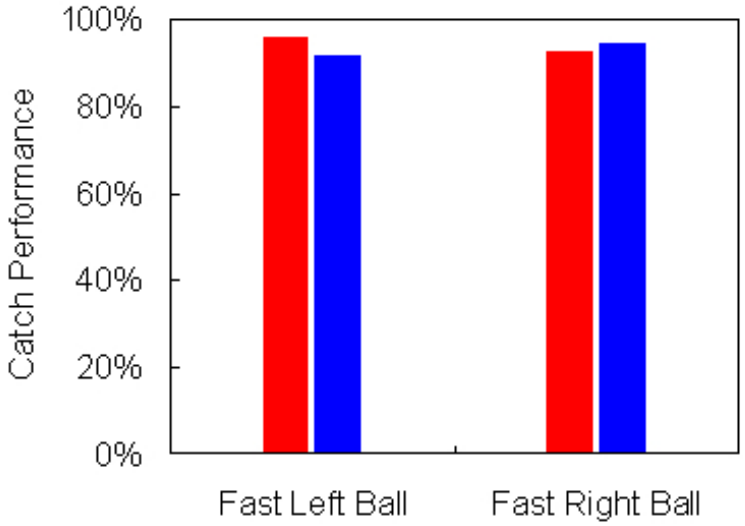
four types of agents were successful in solving the task with a success rate of at least 91.5%. Figure 12 summarizes the results. In the figure, the average performance of the agents under faster left ball and faster right ball conditions are shown. Recurrent network and dropper network both show above 90% performance in all cases (both fast-left and fast-right conditions, and for both high/long [red bars] and low/short capacities [blue bars]).

Between-group performance comparison reveals the recurrent network's slight performance superiority over the dropper network (Figure 13). Nevertheless, what is important here is that the dropper networks show much higher performance than the baseline feedforward networks. Feedforward networks only showed near 50% performance when both fast-left and fast-right conditions are averaged: only catch left ball, catch left and right ball with equal probability, or only catch right ball (data not shown).

How did the feedforward agents using external markers become almost as successful as the recurrent agents? To answer this question, I analyzed the behavior of the feedforward agent to see how they use external markers as a memory-aid. The strategy they used is illustrated in Figure 14. As described in the figure, feedforward agents overshoot the position of the faster-falling balls and then begin to throw external markers. This overshooting patterns of the agents using external markers are clearly observed in their trajectories in Figures 10 and 11. When there is no input from the distance sensors after they catch the ball, they use the external markers in an aversive manner to track back away from the markers until the slower-ball gets



A. Recurrent (LM & SM)



B. Dropper (LS & SS)

Fig. 12. **Average Performance Comparison of the Agents.** The recurrent and the dropper networks show above 90% performance (high capacity/range [red], low capacity/range [blue]).

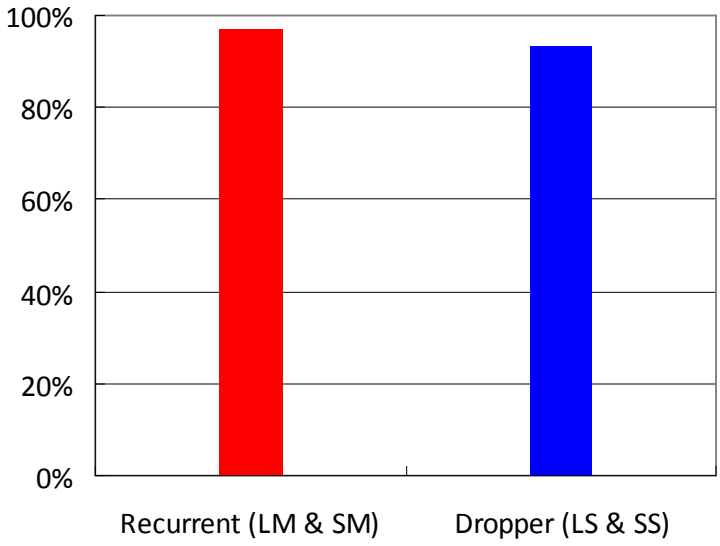


Fig. 13. **Performance Comparison between Recurrent and Dropper Networks.** Average catch performances are plotted. Recurrent networks (M=0.969) show a catch performance greater than dropper networks (M=0.935). The difference between two networks is significant ($p<.001$, $n=4000$).

detected by their distance sensors.

To see how the use of external markers is related to memory, I looked at the hidden state activations of the agents. Figure 15 shows example (LS) of the hidden state activations of the dropper networks throughout the task. We can discriminate the hidden states in "ball-driven" movements from those in "marker-driven" movements. However, the distinction between faster-left-ball and faster-right-ball cases were ambiguous. I further compared the hidden state activations for all four cases (SM, LM, SS, and LS) when the agents are moving back to the second ball after catching the first one because those are the times memory is being used for the recurrent network agents. Figures 16 and 17 plot the hidden state activations in this phase in grayscale

intensities and in line charts.

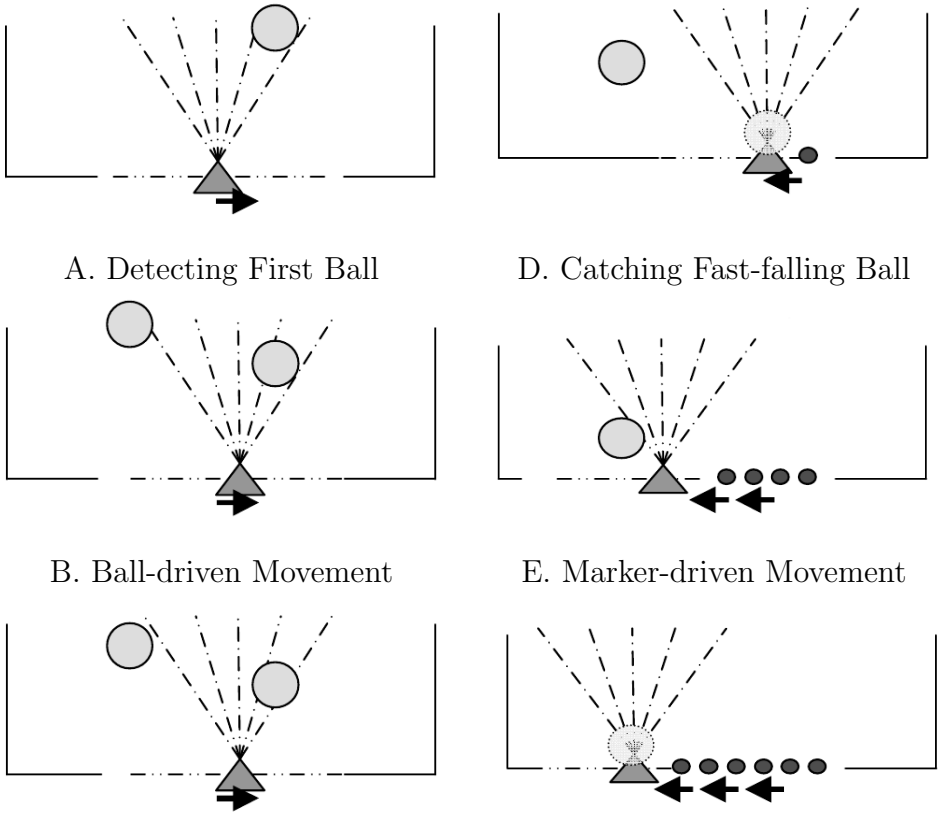
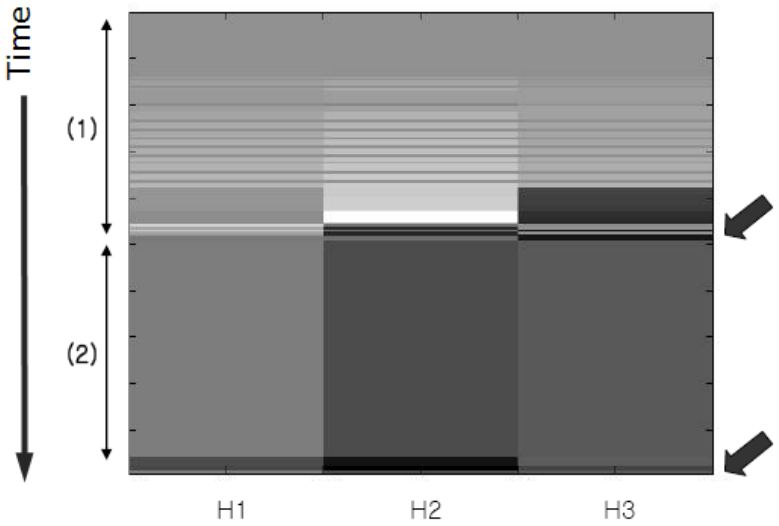
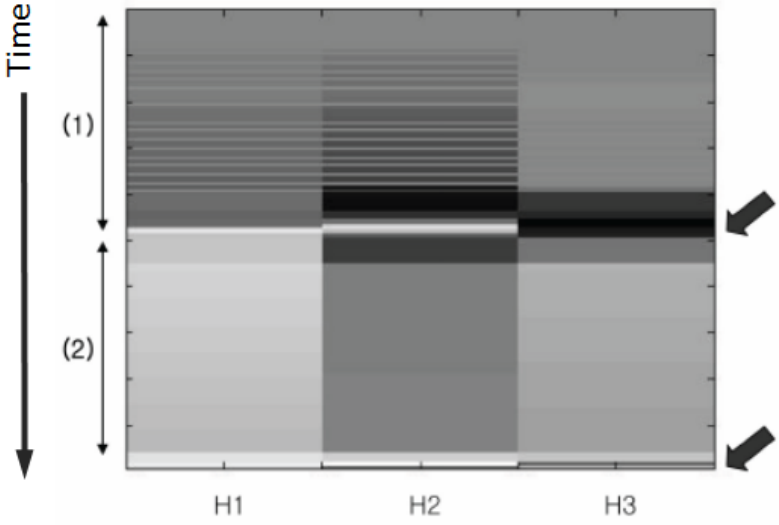


Fig. 14. **Dropper Strategy.** A sketch is shown of a typical evolved strategy observed in dropper agents (SS) in the faster right ball case. A. Agent detects the faster ball first. B. Agent detects both balls while moving toward the faster one. C. The slower ball goes out of the agent's range sensor's scope. D. Agent overshoots the ball and starts to drop markers. E. The markers repel the agent away until the slower ball is detected by the sensor. F. The slower ball comes within the range sensor's scope and the agent catches it.

For the memory-equipped agents using recurrent networks, the difference in the activation patterns of hidden neurons between Case A (faster left ball) and Case B (faster right ball) is evident. For example, in the recurrent network agents (Figures



A. Faster Left Ball



B. Faster Right Ball

Fig. 15. **Example of Hidden States Activation in LS Experiment.** The hidden states for "ball-driven" movements (A-(1) and B-(1)) and for "marker–driven" movements (A-(2) and B-(2)) are distinctive. Ball-catching moments are marked by →.

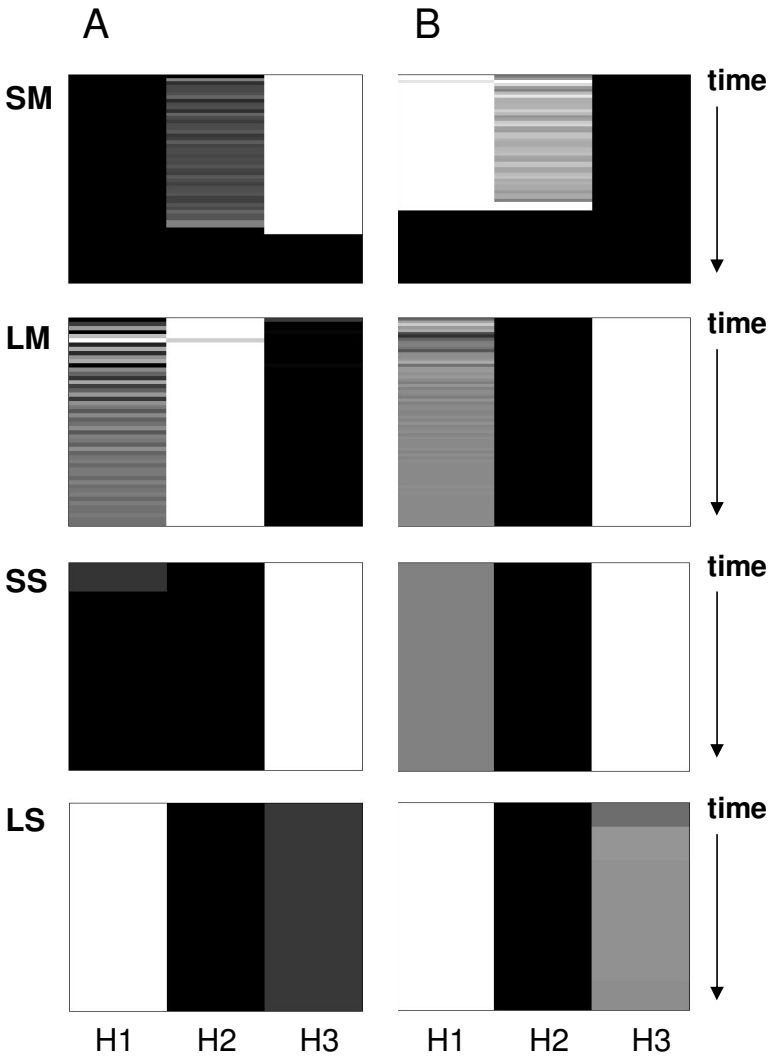


Fig. 16. **Hidden State After First Ball is Caught.** Case A: faster left ball. Case B: faster right ball. SM, LM, SS, and LS mean Short Memory, Long Memory, Short Sensor, Long Sensor, respectively (Table I). H1, H2, and H3 show the hidden layer activation level.

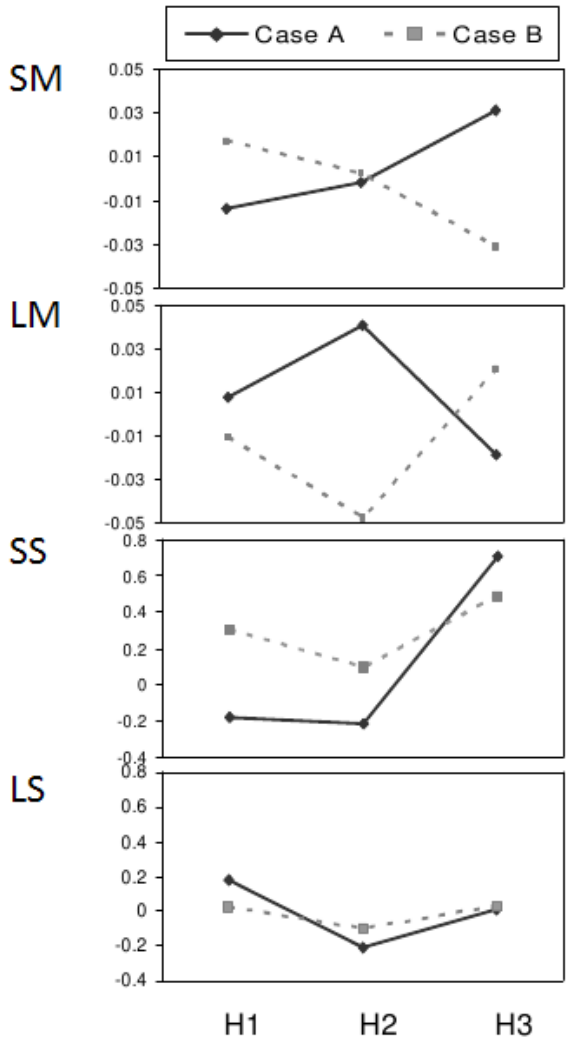


Fig. 17. **Line Chart of the Hidden State After First Ball is Caught.** Line chart of the hidden states in Figure 16. Recurrent networks have distinct internal representations for faster left (Case A) and faster right ball (Case B). However, the two cases are less distinctive in the internal representations of the dropper. SM, LM, SS, and LS mean Short Memory, Long Memory, Short Sensor, Long Sensor, respectively (Table I). H1, H2, and H3 show the hidden layer activation level.

16 and 17, rows LM and SM), the hidden state activation patterns in Case A are near inverse of those in Case B (compare the average activation plot to the right). This hidden unit representation is the internal state of the agent (see e.g. [15]), and since they show distinct states, I can say that the recurrent agents indeed remember the location of the ball (left or right) distinctively. However for the dropper network agents (rows LS and SS), the activation patterns are similar in both Case A and B. This is because the distinction is already presented by the location of the self-dropped markers in the environment (markers on the left-side of the agent in Case A and the right-side of the agent in Case B) and does not need to be represented internally. Thus, the spatial information required to solve the task is located in two different places–inside the brain for the recurrent network or in the environment in the form of external markers for the dropper network.

D. Discussion and Conclusion

In this chapter, I showed how recollection can evolve in neural circuits, thus linking the organism to its past. The results here suggest an interesting linkage between external memory and internalized memory. For example, humans and many other animals use external objects or certain substances excreted into the environment as a means for spatial memory (see [69] for theoretical insights on the benefit of the use of inert matter for cognition). The next chapter (Chapter IV) will extend this model in a more complex memory task of foraging. Because foraging is a behavior critical for the survival of natural agents, this simulation will provide extendibility and biological

plausibility of the dropper network model proposed in this chapter.

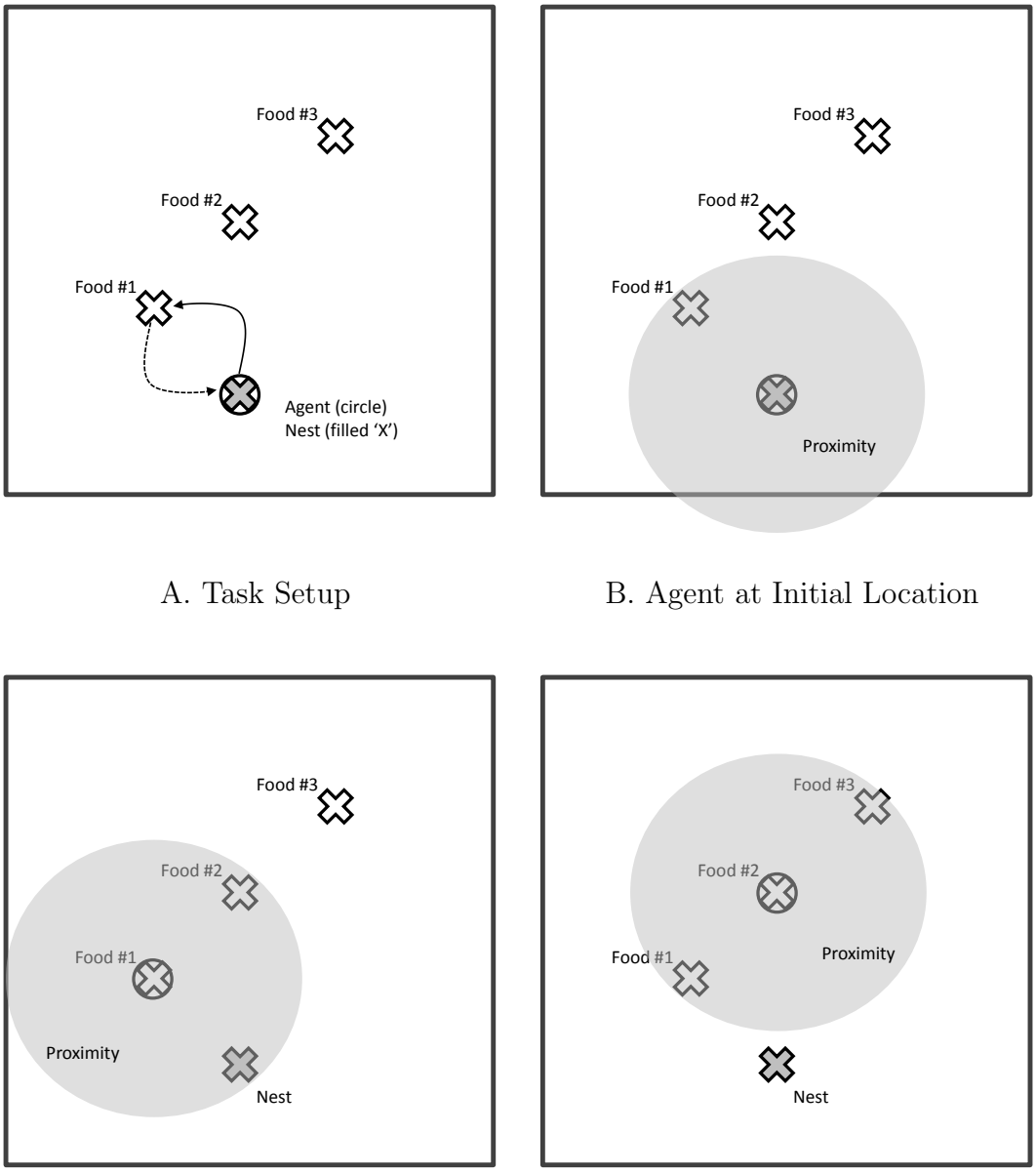
CHAPTER IV

TASK II: FORAGING FOOD IN 2D

To test if the dropper network's performance can be generalized beyond a simple memory task, I extended the task domain to a 2D map. As the task environment becomes more complex, the agent must be able to express richer context. This task was a biologically plausible food-foraging task (Figure 18). Same as with the ball-catching task, I tested recurrent and the dropper networks and compared their performance. I left out the feedforward network since its performance only established the baseline in the previous section.

A. Task Description

Three food items were located in an environment of size 300×300, and the agent starts from the nest location and explores for food. The goal is to consume all three food items in the environment. When a food item is found, the agent needs to find its way back to the nest to consume the food item (The agent has limited food delivery capacity that it can carry only one food at a time). This type of behavior is often observed in animals because they feed the young, cache food for later use, or avoid competition by not consuming the food on-site [70, 71, 72]. The agent has sensors with a limited range of 40, whose inputs are calculated in the same manner as Equation 3.1. The memory requirement of the task is imposed as follows. From the nest, the agent can only detect food#1 (Figure 18B). When it moves close to food#1,



A. Task Setup

B. Agent at Initial Location

C. Agent Getting Food♯1

D. Agent Getting Food♯2

Fig. 18. **Food Foraging Task Definition.** A. Agent's initial location (gray X denotes nest location and blank X's denote food locations). B-D. Shaded area denotes the agent's sensor range. Agent's location is indicated by a circle.

then food#2 comes within its sight (Figure 18C). However, it needs to come back to the nest to consume food#1 and this makes food#2 invisible again. Therefore, the agent needs to memorize where the next food item was (in this example, food#2). The same applies to food#3 (Figure 18D) and the agent has to move back and forth between the food item and the nest at least three times. Moreover, the nest does not generate any sensory cue at all, thus the nest location also needs to be remembered by default. The agent has a limited life span which increases only when it successfully consumes a food item.

B. Methods

The agent model used in this task is illustrated in Figure 19. The agent interacts with the environment with 8 pairs of sensors distributed uniformly around the center at an interval of 45°, 8 of which are receptors sensitive to the external markers the agent drops. The other 8 are distance sensors sensitive to food items. Both types of sensors are limited in range subject to the constraint mentioned in the foraging task description, and the magnitude of the sensor signal is inversely proportional to the distance to the detected object.

Recurrent and dropper networks used in this task have similar topologies as described in the ball-catching task, with minor differences in the number of input and output units. Three network outputs (O_1, O_2, and O_3) indicate x-, y-, and orientation offsets, which determine the next location and orientation of the agents:

$$\theta(t+1) = \theta(t) + 180^\circ \times O_3(t) \tag{4.1}$$

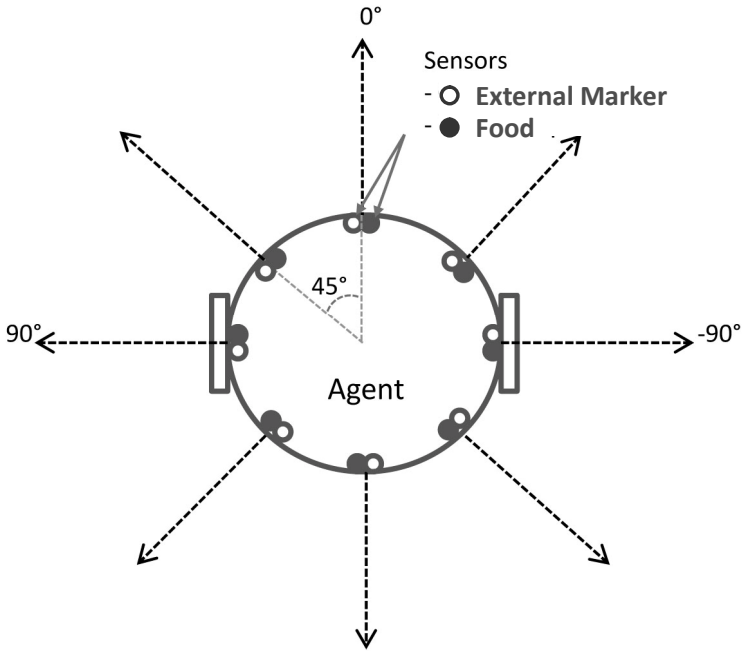


Fig. 19. **Agent Model.** Simple agent with directional/ranged sensors for food and external markers.

$$\begin{aligned} x(t+1) = x(t) \\ &+ Speed \times O_1(t) \times \cos(\theta(t+1)) \\ &+ Speed \times O_2(t) \times \cos(\theta(t+1) + 90^\circ) \end{aligned} \tag{4.2}$$

$$\begin{aligned} y(t+1) = y(t) \\ &+ Speed \times O_1(t) \times \sin(\theta(t+1)) \\ &+ Speed \times O_2(t) \times \sin(\theta(t+1) + 90^\circ) \end{aligned} \tag{4.3}$$

where $\theta(t)$, $x(t)$, and $y(t)$ are the orientation and xy location of the agent at time t (Figure 20). *Speed*, the speed of the agent, was set to 3. The agent can freely navigate within the environment while bouncing off when it comes into contact with the wall (Figure 21). As in the previous task, default movements are blocked by removing the

bias units so that the agent does not move without sensory input. This is intended to maximize the memory requirement of the task. Also, one more output unit is added to a typical feedforward network to allow marker dropping behavior (Figure 22). To analyze the effect of 'forgetfulness' of the memory, three different memory decay rates were tested for both the recurrent and the dropper networks. Table II summarizes the experimental setup. I tried three different memory decay rates of the recurrent network (λ=1.0, 0.99, and 0.7). For each of them, three previous hidden state vector sizes (N_{mem}=5, 10, and 20, see Equation 3.3 and Figure 8) were tested to identify their effects on the performance. The evaporation rate (ρ) of the chemical markers used by the dropper network was also varied as the memory decay rate of the recurrent networks: $M(t) = \rho M(t-1)$, where ρ=1.0, 0.99, and 0.7, and $M(t)$ is the marker strength at time t. Volatility is a natural property of chemicals analogous to the recency effect (forgetfulness) of internal memory. If so, using this property can increase performance by prioritizing newer events, as the recency effect does [73]. The same genetic algorithm discussed in the above section was used to train the networks.

Table II. Food Foraging Task Experiment Description

Memory Decay	Agent		
	Recurrent		Dropper
0% loss per step	λ=1.0	N_{mem}=5 N_{mem}=10 N_{mem}=20	ρ=1.0
1% loss per step	λ=0.99	N_{mem}=5 N_{mem}=10 N_{mem}=20	ρ=0.99
30% loss per step	λ=0.7	N_{mem}=5 N_{mem}=10 N_{mem}=20	ρ=0.7

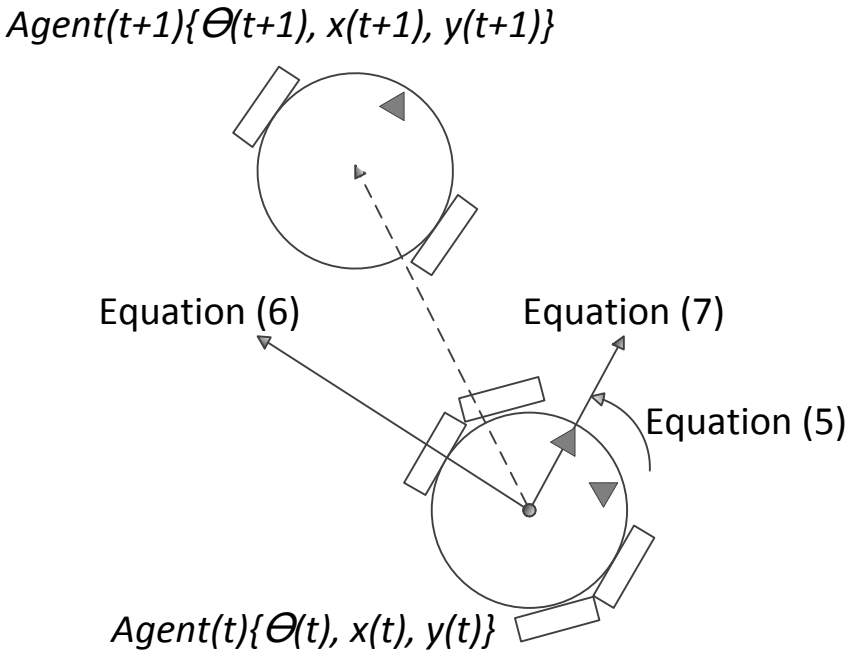


Fig. 20. **Agent Movement.** *Agent*(t) denotes the agent's orientation and location at time t. It turns and moves according to Equation(5), (6), and (7).

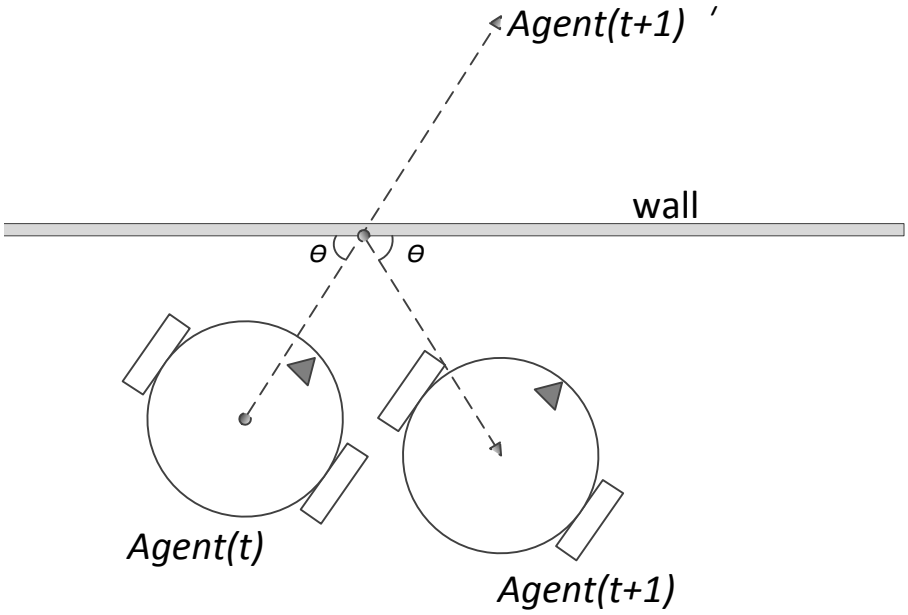


Fig. 21. **Agent Bounces off the Wall.** $Agent(t+1)'$ denotes the agent's next location calculated using Equation (5), (6), and (7). As the agent contacts a wall on its way to $Agent(t+1)'$, it bounces off the wall and the next location is changed to $Agent(t+1)$.

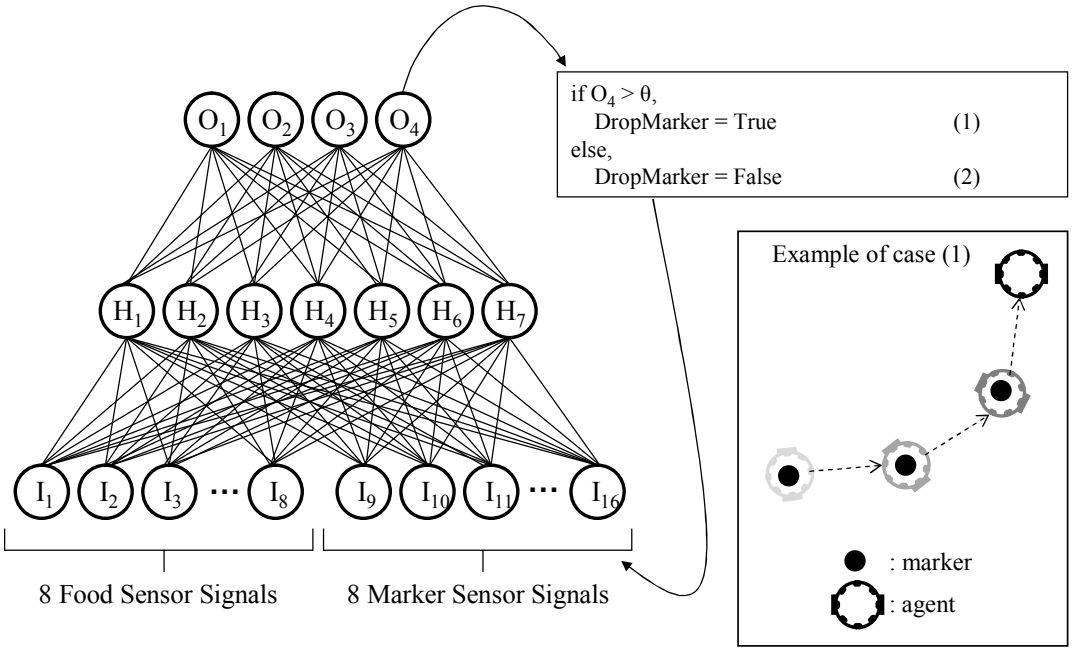


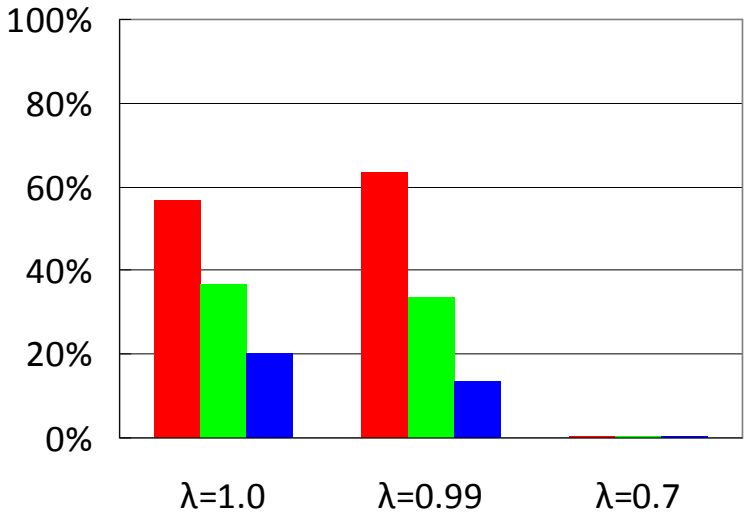
Fig. 22. **Neural Network of the Dropper Agent.** An output unit (O_4) is added to the typical feedforward network to allow external marker dropping behavior. The box shows an example of leaving external markers when the agent moves to a next location. The recurrent network counterpart had a similar structure as Figure 8, with no external marker sensors.

C. Experiments and Results

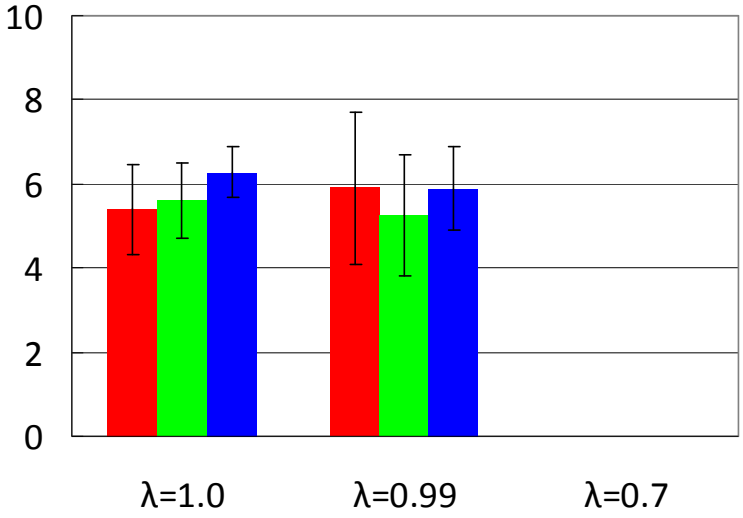
A total of 30 evolutionary trials were given to each type of network controller. A trial is recorded as successful if the agent consumes all three food items in the environment. Figures 23 and 24 show the quantitative results of the experiment. It is clear that the growing number of hidden state feedback degrades the performance of the recurrent networks (Figure 23A). This is because of the additional parameters that the recurrent agents have to tune. For example, the recurrent agent with 5 hidden state feedbacks has 8×7 (input to hidden weights)+5×7×7 (hidden feed-

back weights)+7×3 (hidden to output weights)=322 parameters while the dropper agent has only 16×7 (input to hidden weights)+7×4 (hidden to output weights)+1 (threshold for O_4)=141 parameters. When the number of hidden state feedback is 20 for the recurrent agent, the number of parameters grows to 1057. Comparing the success rates between λ=1.0 and λ=0.99, small memory decay did not seem to have a significant impact for the performances of the recurrent networks. Only the success rate of the one with 5 hidden states feedbacks slightly grew as λ changes from 1.0 to 0.99. Inverse correlation is observed between the success rate and the travel distance when $\lambda = 1.0$. However, such correlation is not observed when $\lambda = 0.7$ (Figure 23B). When the memory decay was 30% per step, no recurrent network was successful. On the other hand, the performances of the dropper networks grew once I let the markers evaporate (Figure 24). The dropper networks show fairly high (80%) success rate even with high marker evaporation rate (ρ=0.7). No agent with random network weights in either network type (recurrent and dropper) was found successful (data not shown). For the traveled distances, the recurrent networks moved longer distances than the dropper networks, regardless of the different memory decay rate (Figures 23B and 24B). When the evaporation rate changed from ρ=0.99 to ρ=0.7, the travel distance of the dropper network increased and fewer markers were dropped. Further qualitative analysis elucidates the quantitative differences.

Figures 25 and 26 plot the traces of successful recurrent agents with two different memory decay rates (λ=1.0 and 0.99). The agents do not have explicit knowledge about the nest location, because the nest does not radiate any information and the

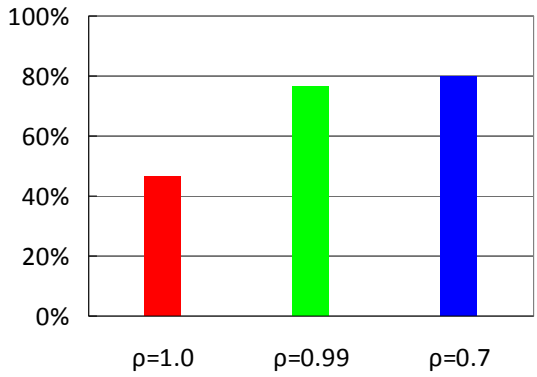


A. Success Rate

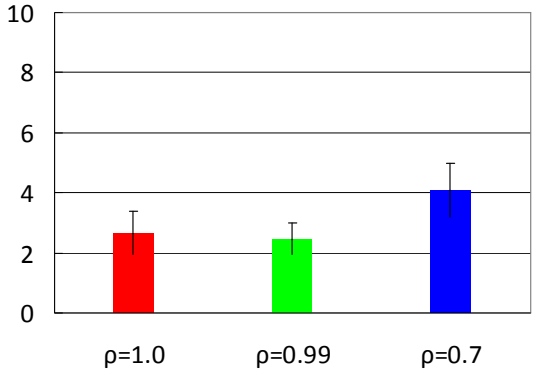


B. Distance Traveled

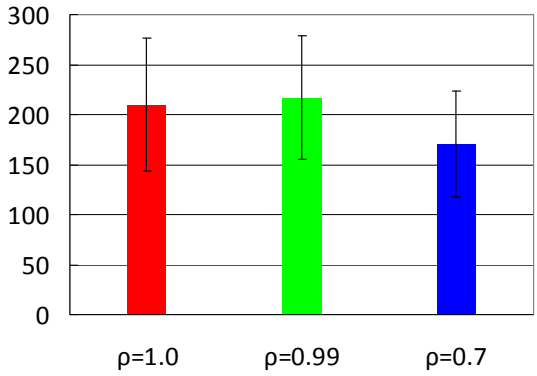
Fig. 23. **Quantitative Data of the Successful Recurrent Networks.** Colors indicate different sizes of the hidden state feedback (red: N_{mem}=5; green: N_{mem}=10; blue: N_{mem}=20, see Equation 3.3 and Figure 8). A. Performance degrades as the number of hidden states feedback grows. B. Distance metric is a ratio of the distance traveled and the minimum distance needed to reach a goal. Overall, the succesful recurrent networks traveled long distances (min avg.=5.249 with λ=0.99, N_{mem}=10; max avg.=6.277 with λ=1.0, N_{mem}=20).



A. Success Rate



B. Distance Traveled



C. Marker Drops

Fig. 24. **Quantitative Data of Successful Dropper Agents.** A. Large difference in success rate is observed once markers evaporate (14/30 for ρ=1.0; 23/30 for ρ=0.99; 24/30 for ρ=0.7). B and C. Agent moved significantly longer distance and dropped less number of chemical markers when evaporation rate was the highest (ρ=0.7). For the distance metric used in B, see Figure 23B.

agents do not have a mechanism to change their states based on whether they visited the nest or not. Moreover, the hidden state feedbacks affect the movements of the recurrent agents, regardless of their current locations. Therefore, it is difficult for the recurrent agents to change their states to turn abruptly at the nest location. Under this condition, the fittest strategy found by artificial evolution for the recurrent agents was a repeated circular movement. Even though this strategy enabled the agent to be successful in the task, the qualities of the solutions were poor, as can be seen from the behavioral trajectories shown in Figures 25 and 26. As we can see in these plots, the agents seemed to blindly scan through the task arena repeatedly drawing circles. The solutions they found almost seemed to be out of pure luck, since the trajectories do not show any goal-directedness. As a result, the orders of the food items found by this strategy were often different from the order that the food items were presented (1-2-3). For example, the order that the food items were found in the 4th panel of Figure 25 and in the 2nd panel of Figure 26 is 2-1-3.

Figures 27, 28, and 29 plot the traces of the dropper agents with three different marker evaporation rates ($\rho = 1.0$, 0.99, and 0.7 respectively). For the dropper agents, this task is not too difficult because the markers explicitly contain spatial information. For example, after foraging a food item, the dropper agents are guaranteed to get to the nest just by following the marker trail. Also, the memory source is local and thus dependent on the current location of the agents because they can sense only the markers within their sensor radius. Regardless of their current location and orientation, the dropper agents only need to evolve their connection weights between

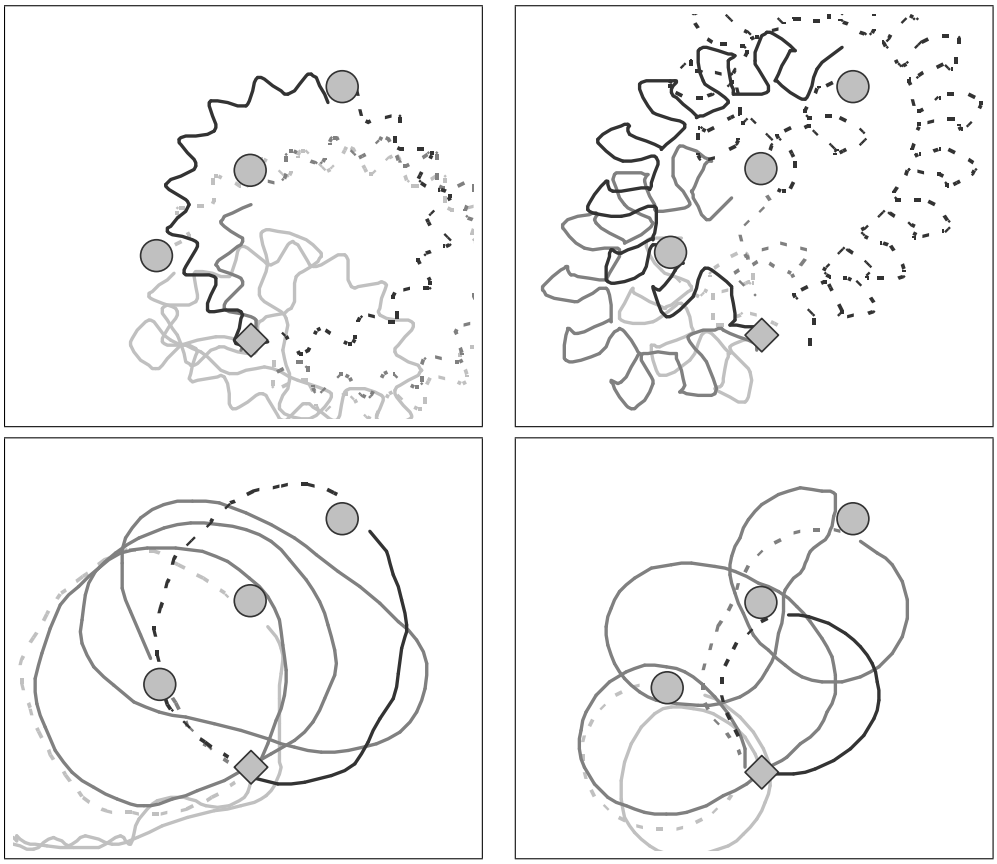


Fig. 25. **Trajectories of Successful Recurrent Agents with $\lambda = 1.0$.** Agents are moving in circles to get to the food items and the nest. In general, the trajectories do not show any goal-directedness (move directly toward food or toward nest). The gray diamonds at the bottom of each panel indicate the nest and the gray circles denote food items. The grayscale intensity of the line denotes the temporal order of each trajectory (the darker, the more recent). The line type (solid/dashed) distinguishes between the nest-to-food/food-to-nest trajectories.

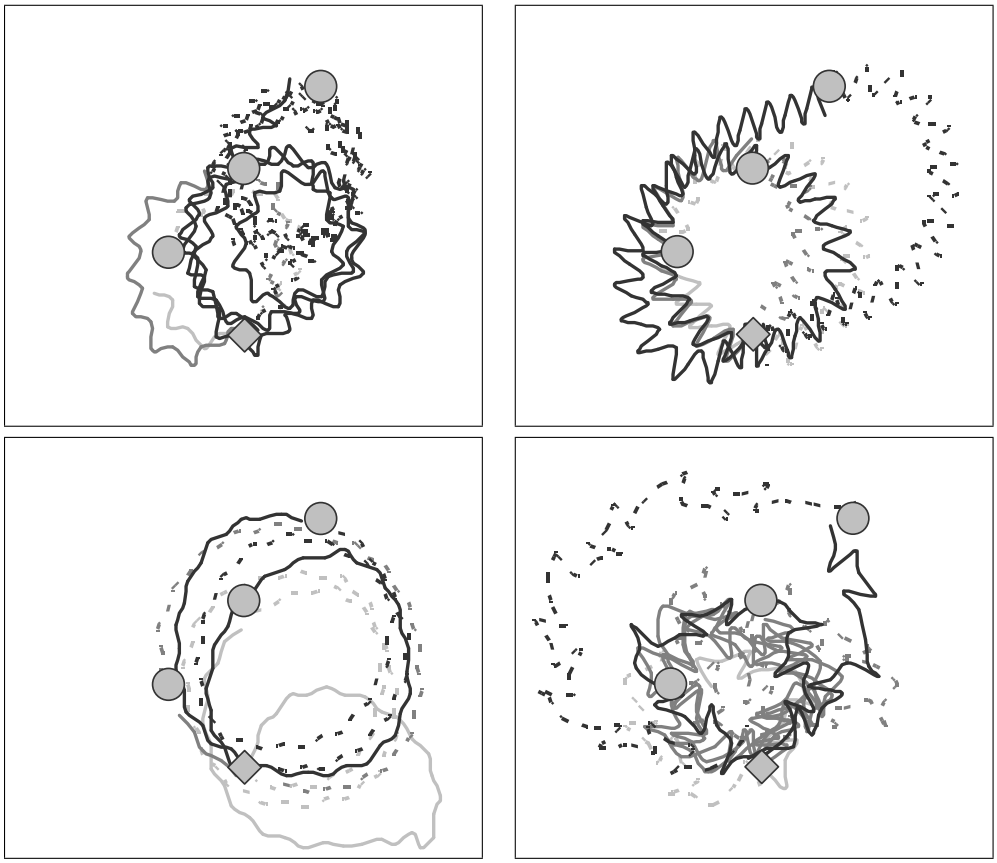


Fig. 26. **Trajectories of Successful Recurrent Agents with $\lambda = 0.99$.** Similar to the case with $\lambda = 1.0$, agents are again moving in circles without goal-directedness. See Figure 25 for plotting conventions.

their sensors and actuators (input and output neurons) to generate movement so that they can maintain the distance to keep some of the markers still visible at the next location. As a result, only few repeating circular movements are observed, and the dropper agents usually came back to the nest with a food item once they were out for foraging. Compared to the trajectories of recurrent networks (Figures 25 and 26), the trajectories of the dropper agents are more goal-directed and this is why the average travel distance of the dropper agents is much shorter than that of the recurrent agents. When ρ=0.7, the markers evaporate too fast and the dropper agents can use the most recent markers only (Figure 29). The limited marker information made the dropper agents to generate wiggling trajectories, thus requiring longer travel distance. To fully account for the effect of evaporating markers, I compared the dropper agents with ρ=1.0 (Figure 27) and ρ=0.99 (Figure 28). These two ρ values were selected because their qualitative results showed the most difference.

Even though both types of dropper agents (learned similar circular movement in general, the dropper agents in the E group show much more compact trajectories (Figures 27, 28, and 29), and this can be the explanation for the difference in the travel distance in Figure 24. This compact trajectories made the markers dropped closer to the locations of importance. Figure 30 plots the result of k-means clustering (k=4) applied to the set of all markers dropped by successful dropper agents in both groups. Clusters of non-evaporating markers do not match the food items and nest locations, whereas evaporating markers are dropped in correspondence with these locations. To fully account for the reason for such differences in trajectories and the number of

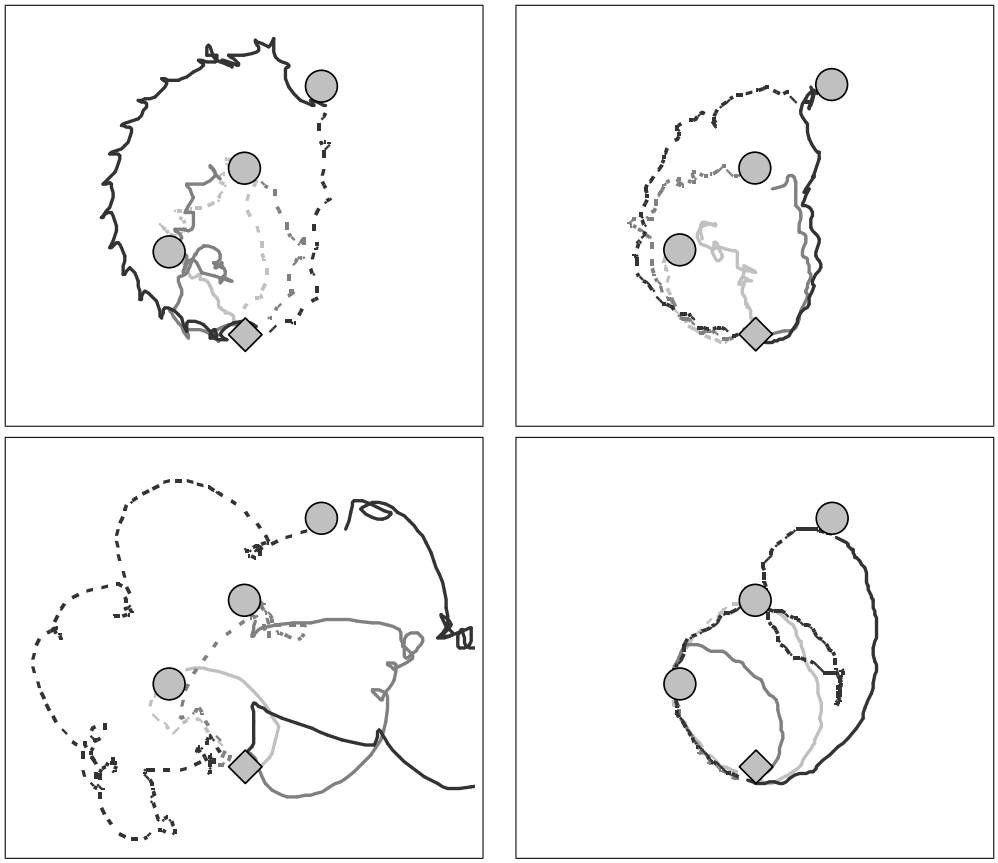


Fig. 27. **Trajectories of the Successful Dropper Agents with $\rho = 1.0$.** When the markers do not evaporate (ρ =1.0), the dropper networks move in circles with growing radius. See Figure 25 for plotting conventions.

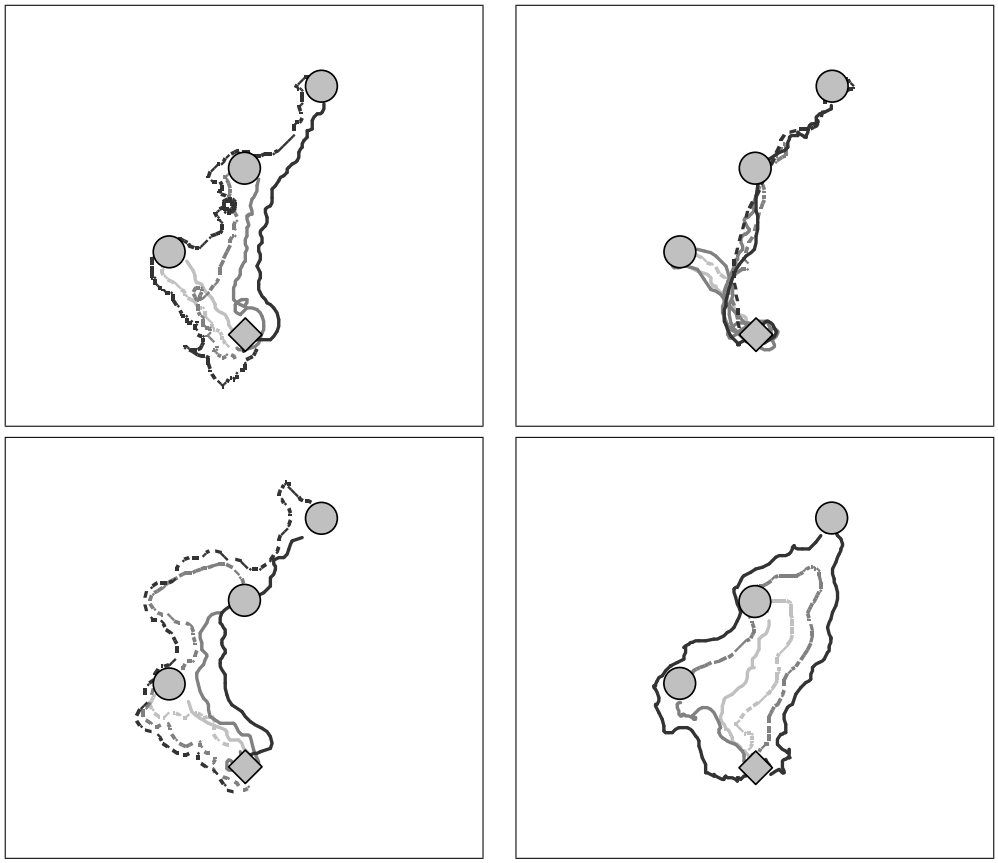


Fig. 28. **Trajectories of the Successful Dropper Agents with $\rho = 0.99$.** Temporal information in evaporative markers (ρ =0.99) enabled the dropper agents to generate very compact trajectories. See Figure 25 for plotting conventions.

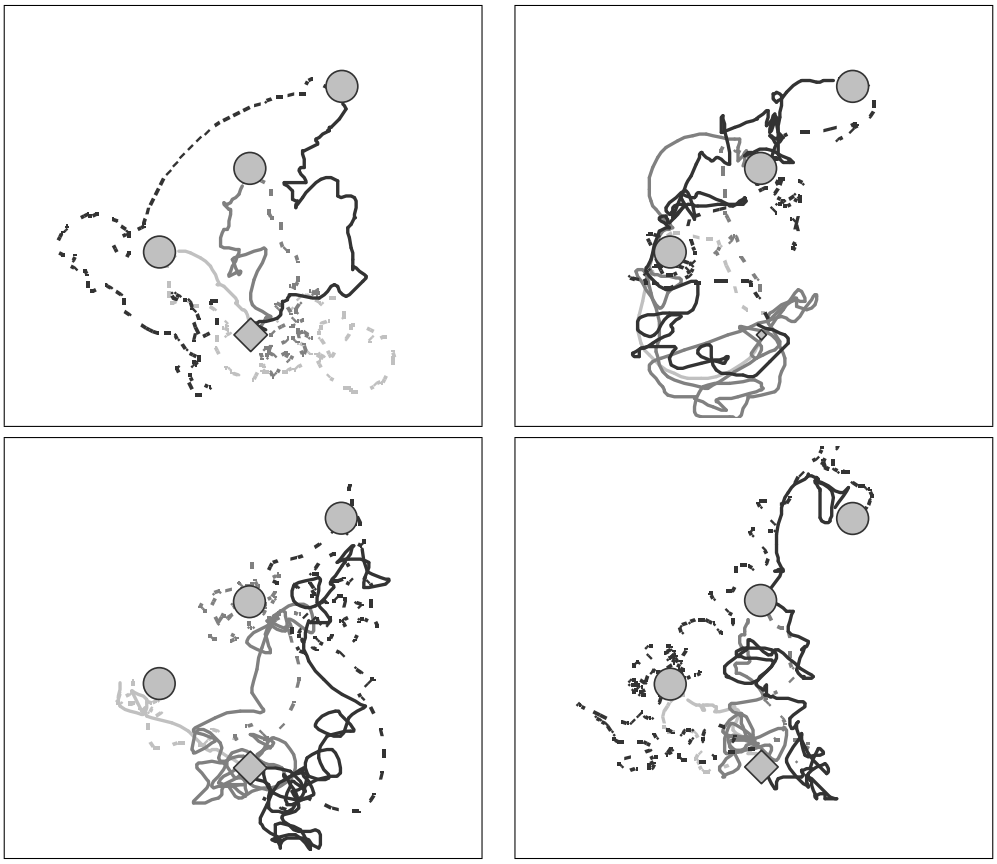


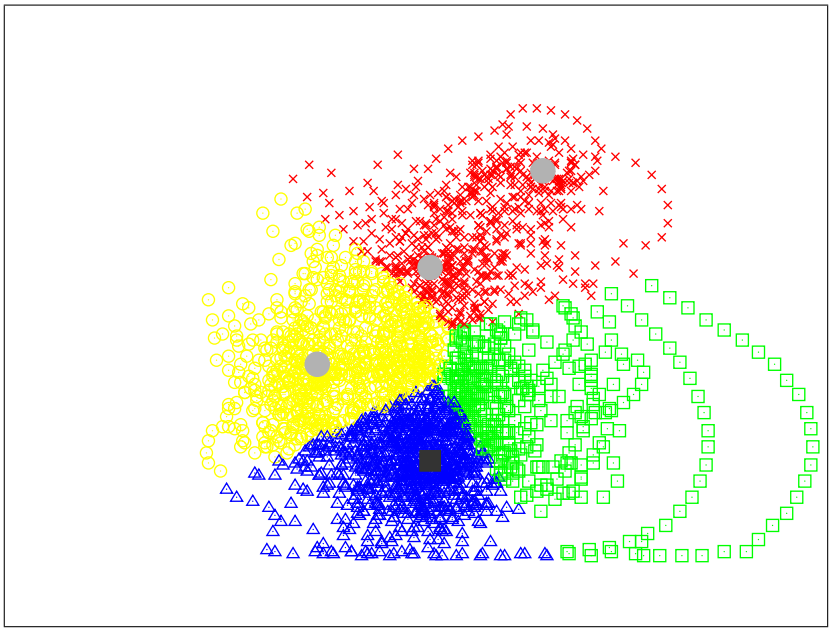
Fig. 29. **Trajectories of the Successful Dropper Agents with $\rho = 0.7$.** The dropper agents show 'wiggling' trajectories with greater memory decay (ρ =0.7). As the markers are too volatile, the dropper agents can utilize the most recent markers only. As a result, they end up with extensive search of the environment. See Figure 25 for plotting conventions.

markers thrown, I performed a detailed comparison between the best dropper agents in each group. Dropper agents with the shortest step size in both groups were selected as being the best because they have the most economical strategy among their group members in solving the same task.

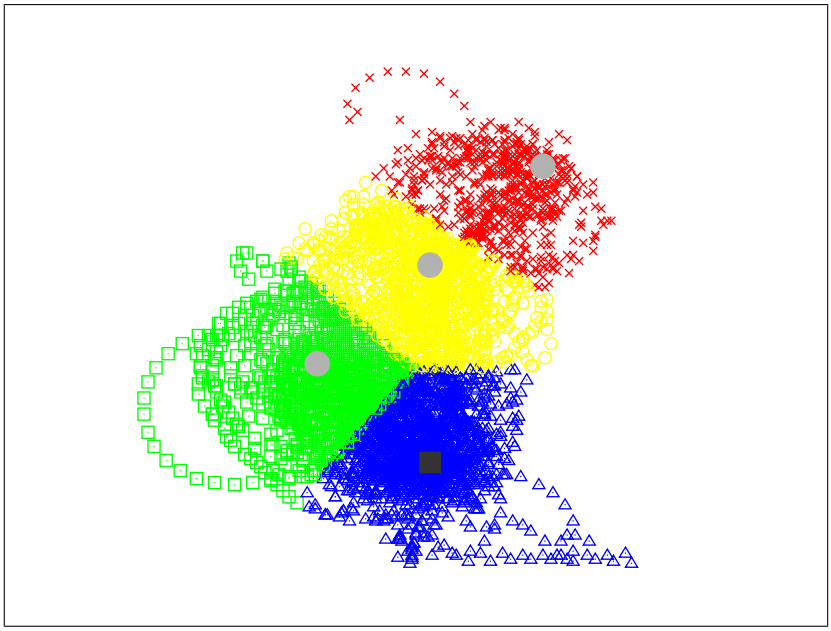
One-to-one comparisons of the strategy and the trajectory between the best-performing agents in each marker evaporation rate show divergence in the evolved behaviors. Figures 31 and 32 confirmed more compact trajectory developed by agents using evaporating markers (Figure 31-(1) vs. Figure 32-(1)). (2) and (3) of Figure 32 show the evaporation of chemical markers in fading color. As the chemicals evaporate, their sensitivity fades and the attraction toward the non-existing (already-consumed) food becomes weaker and less competitive than the attraction to the remaining food. Because of the difference in the volatile characteristic of markers, the two groups developed different strategies for the task. The strategies are summerized below:

- Dropper networks using non-evaporating markers (ρ=1.0)

    1. Approach food item from right to left (Figure 31-(1)).
    2. Follow markers laid on its left side (Figure 31-(1)).
    3. If food item is detected,
        (a) If marker is detected rather far, follow the food item while throwing markers (Figure 31-(2)).
        (b) If marker is detected very near, follow the marker without throwing another one (Figure 31-(3)).



A. Non-Evaporating Chemical



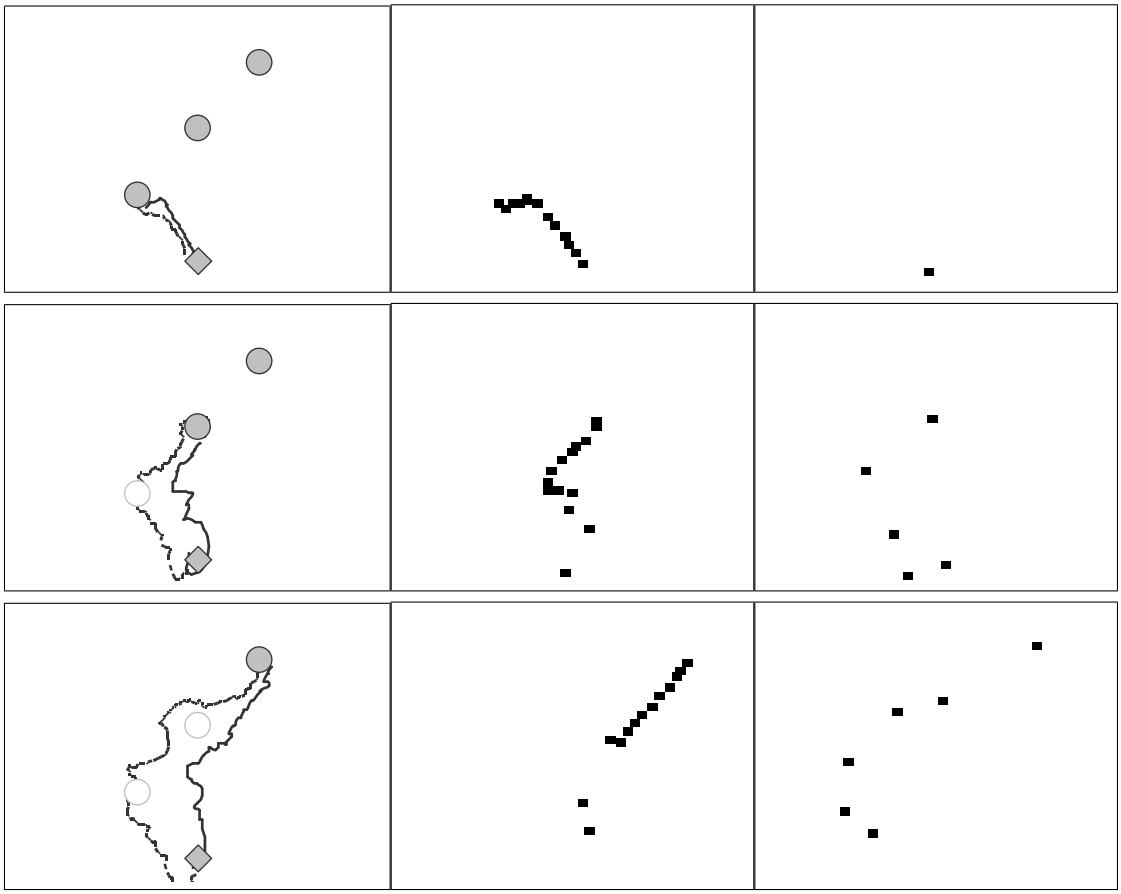
B. Evaporating Chemical

Fig. 30. **Cluster of Chemical Markers.** K-means clustering (k=4) is applied to the chemical markers to show the relationship between the markers and points necessary to remember for the task (food&nest).

- Dropper networks using evaporating markers (ρ=0.99)

    1. Always throw markers (Figure 32-(1)).
    2. If food item is detected,
        (a) If marker signal is week (far or old), follow the food item (Figure 32-(2)).
        (b) If marker signal is strong (close and recent), follow the marker (Figure 32-(3)).

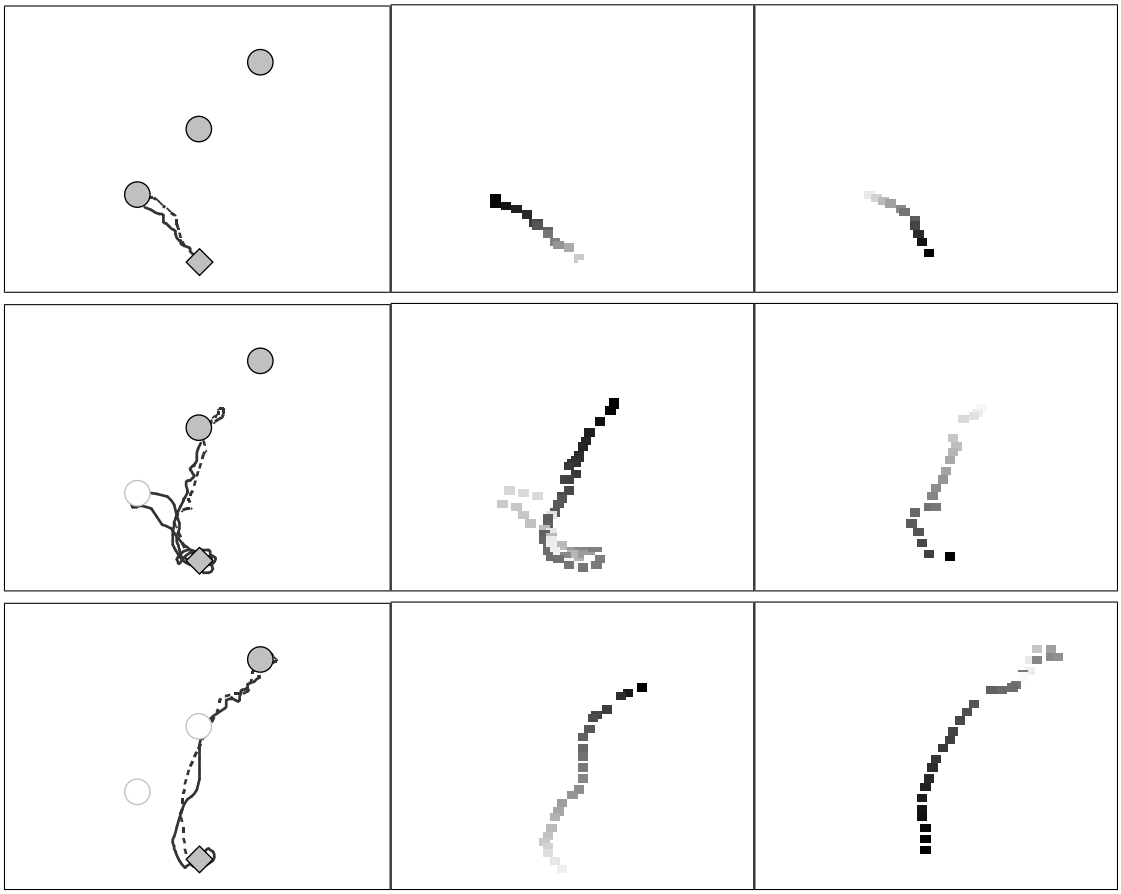
The strategies laid out above show that agents using non-evaporating markers (ρ=1.0) evolved additional rules to drop/detect markers in a particular direction (left side in this example), whereas the agents using evaporative markers has simpler drop/detect rule. This is due to the evaporating property of the markers. Because the evaporative markers contain temporal information, it does not have to develop additional rules to avoid conflicts among the markers. As the markers evaporate, their sensitivity fades and the attraction toward the non-existing (already-consumed) food item becomes weaker than the attraction to the remaining food item. However, because the non-evaporating markers cannot encode the passage of time, a tie-breaking function for conflicting markers had to be developed by the agent. To summarize, utilizing self-generated olfactory-like cues enables the agent to exploit the volatility of markers, a natural property of chemical markers. As a result, the dropper agent can distribute, in the environment, information necessary to solve a task, and its interaction with the environment enables it to communicate with itself in the past.



(1) (2) (3)

No Evaporation (ρ=1.0)

Fig. 31. **Agent Trajectory and Non-evaporative Chemical Trails (solid: nest–to-food, dashed: food-to-nest trajectory).** Agents using non-evaporating chemical markers developed more complex set of rules.



(1) (2) (3)

Evaporation (ρ=0.99)

Fig. 32. **Agent Trajectory and Evaporative Chemical Trails (solid: nest–to-food, dashed: food-to-nest trajectory).** Agents using evaporative chemical markers developed simpler set of rules.

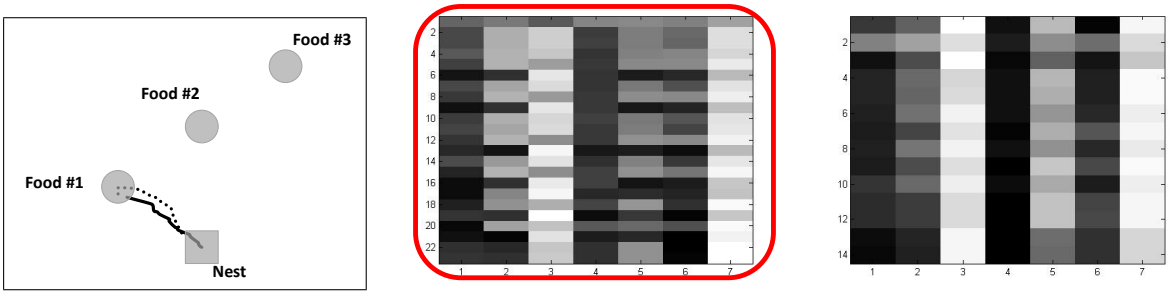
This kind of *stigmergy* lowers the complexity of the solution.

I again took a look at the dropper agent's hidden state activations. Figure 33 shows the hidden state activations of the best agent using evaporative markers. We can observe significantly more fluctuations in the hidden states activations when the agent is generating "food-driven" movement (red-marked area). I measured the degree of fluctuation using a simple metric:

$$Fluctuation = \sum_{i=2}^{N_t} \sum_{j=1}^{N_{\text{hid}}} \text{abs}(H_{i,j} - H_{i-1,j})/N_t \tag{4.4}$$

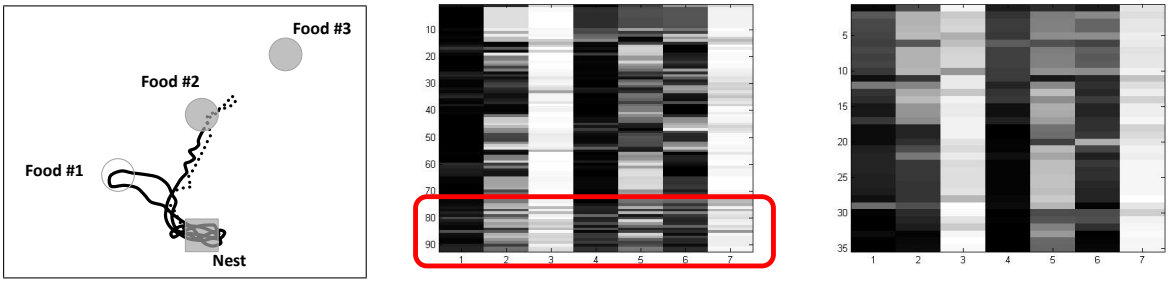
where $H_{i,j}$ is the activation value of j^{th} hidden neuron at time i, N_t is the maximum step size, and N_{hid} is the number of hidden neurons. abs$(\cdot)$ is the absolute value function. The difference in the fluctuation amount in "food-driven" and "marker-driven" movements is shown in Figure 34. To see this difference from a different perspective, Figure 33C-(2) was replotted as a line chart (Figure 34B). Even though it is hard to tell the details of the detected markers, the agent's internal states can show visible distinction during different behaviors ("food-driven" vs. "marker-driven"), which is similar to the previous result in the ball-catching task.

An interesting insight arises when I compare the external and internal implementation of memory. External memory is more persistent, while internal memory is more transient due to its dynamic nature so each may have specialized in the respective kinds of tasks. Probably this is why both systems are preserved in modern animals. Therefore, it would be interesting to observe the evolution of hybrid networks with both recurrent connection and marker dropping/detecting ability. In fact, the results



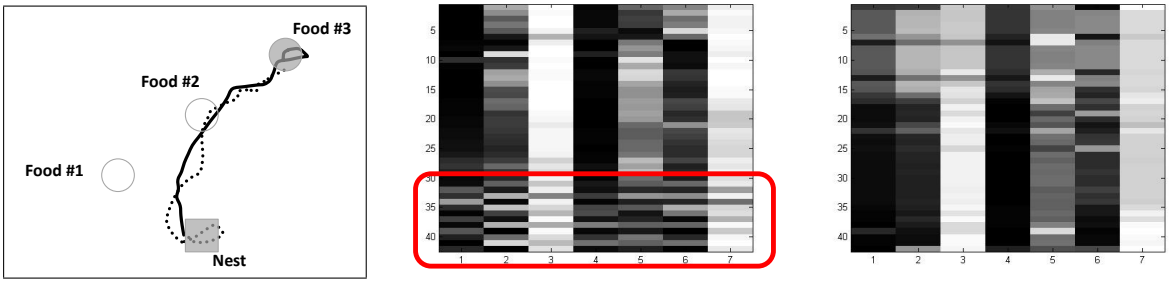
(1) Agent Trajectory (2) Hidden States (toward food) (3) Hidden States (toward nest)

A. Foraging Food ♯1



(1) Agent Trajectory (2) Hidden States (toward food) (3) Hidden States (toward nest)

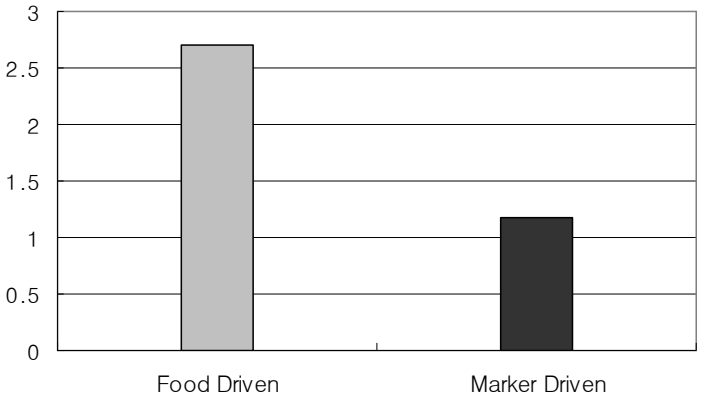
B. Foraging Food ♯2



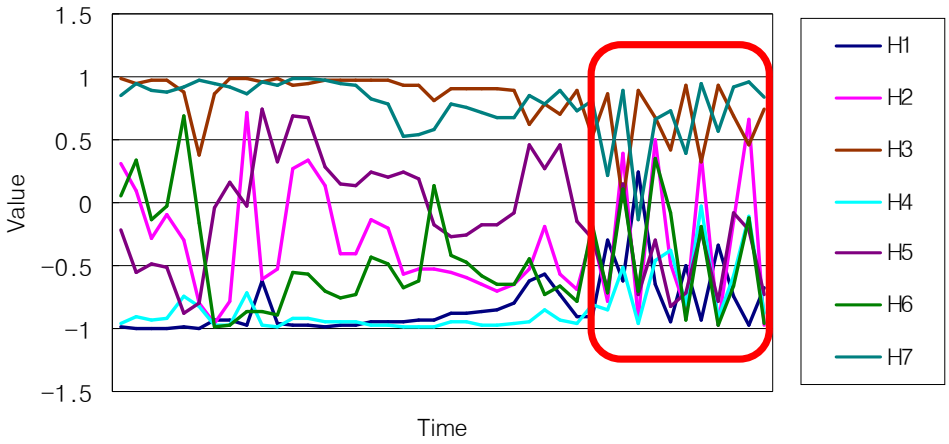
(1) Agent Trajectory (2) Hidden States (toward food) (3) Hidden States (toward nest)

C. Foraging Food ♯3

Fig. 33. **Hidden State Activations.** (1) Same as Figure 32. (2) Areas bounded by the red box denote the hidden state activation during "food-driven" movements, where hidden states fluctuate more vigorously.



A. Fluctuation Difference

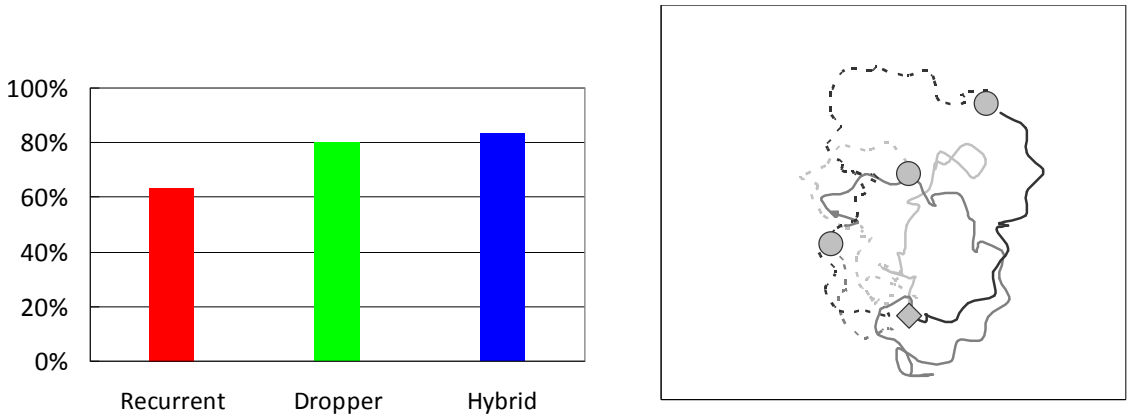


B. Different Perspective of Figure 33C-(2)

Fig. 34. **Difference in Hidden State Activation.** A. Amount of fluctuation is calculated according to the equation 3.4. B. Line chart of Figure 33C-(2). Area marked in red denotes the same area in Figure 33C-(2).

suggest that a hybrid can outperform the non-hybrid versions (Figure 35). Note that the performances of the recurrent and the dropper networks are not directly comparable because the parameters used in each network type were arbitrarily chosen and not optimal. However, it is meaningful to compare the recurrent or the dropper to the hybrid network to see how one type of memory is reinforced by the other, and it will be an intriguing future work to analyze if there exists a certain condition that makes one type of memory (external or internal) more preferable to the other. Finally, the low performance of recurrent agents may also be due to the difficulty of evolving networks with a huge number of tunable parameters (the connection weights). The dropper network is architecturally almost the same as the feedforward network, while the function is much more advanced. This kind of economy could surely have been exploited by the process of evolution.

One possible criticism is that the tasks themselves do not require elaborate spatial memory, and can be tackled with taxon navigation. Taxon navigation uses a reactive stimulus-response strategy, with which agents generate direct homing onto landmarks. Taxon navigation requires a visible landmark at all times, whereas in my tasks, due to the limited sensor range, landmarks (or objects other than the self-generated markers) are not always visible, thus the task itself requires spatial memory. However, it is still possible that my agents used taxon navigation since all three food items were always at the same location throughout trials. To exclude this possibility, I evolved my dropper agent by varying the food item locations. The initial results were encouraging (Figure 36), where the same evolved agent was able to solve all four different food



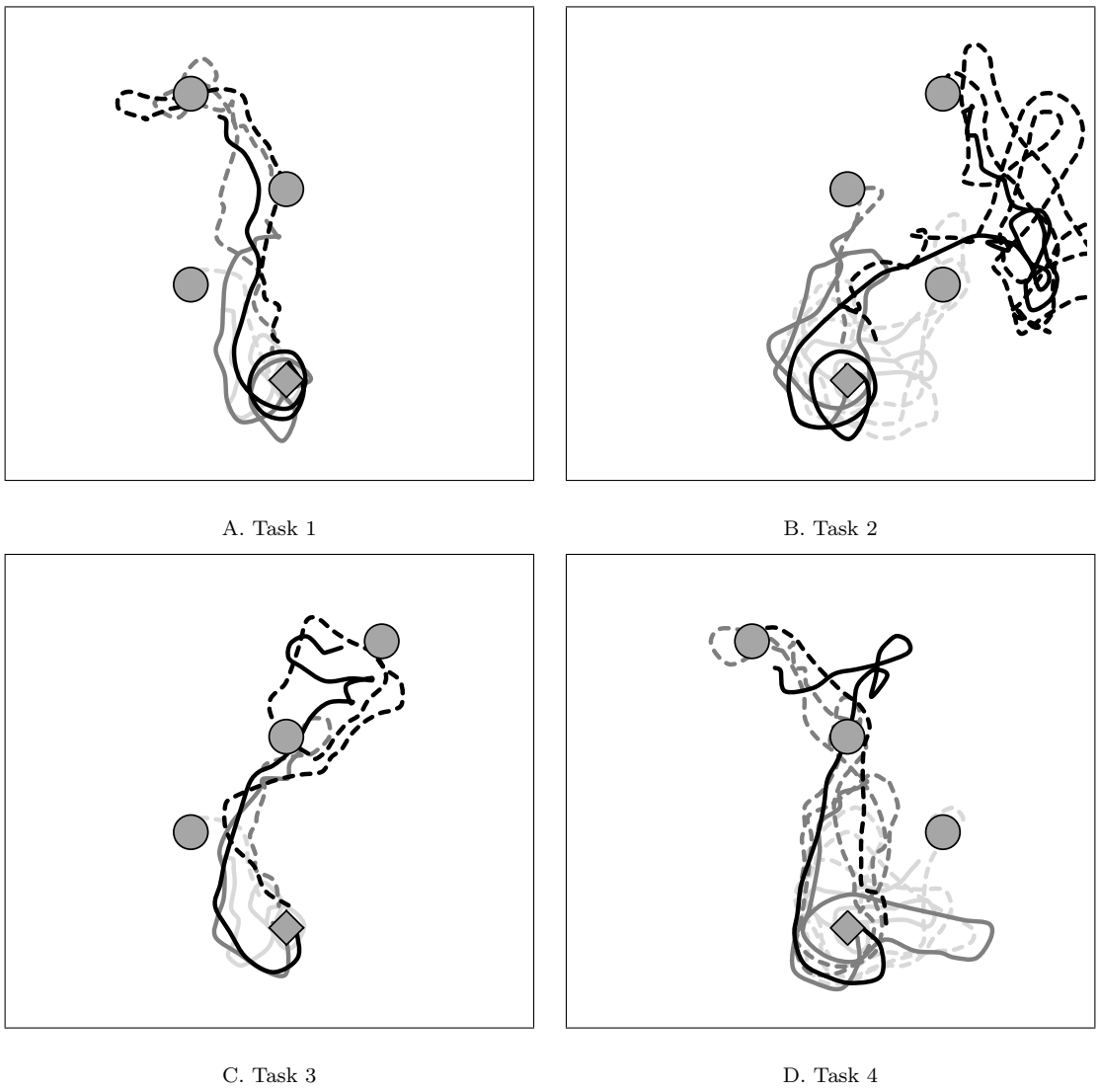
A. Success Rate B. Sample Trajectory

Fig. 35. **Preliminary Result of Hybrid Networks.** A. Success rate comparison. Colors indicate different network types (red: recurrent; green: dropper; blue: hybrid). The best performing configuration of each network is shown (recurrent: λ=0.99, N_{mem}=5; dropper: ρ=0.7; hybrid: λ=0.99, N_{mem}=5, ρ=0.7). See Figures 23 and 24 for details. B. Trajectory of a sampled hybrid network. See Figure 25 for plotting conventions.

item configurations, thus showing the agent's spatial memory function. The reactive part of the agent indeed employs a taxon navigation strategy, but the landmarks in this case are both food items and agent-generated markers. Thus, as a whole (i.e., including the marker dropper/sensing capability), the agent cannot be said to be following a taxon-based strategy.

D. Discussion and Conclusion

Combined with the results in the previous chapter (Chapter III), I have shown that reactive feedforward neural network can exhibit behavior requiring memory when coupled with a simple external material interaction mechanism. It is surprising that feedforward networks can decide when to drop the external marker, and what to do when a marker is detected, thus using the environment as an open canvas. This external material interaction mechanism could have indeed been used by memoryless basal animals to increase their survival. I strongly believe such external material interaction is a necessary intermediate step for the evolution of memory, from external to fully internal. The following chapter (Chapter V) will examine that passing through such an intermediate step has evolutionary advantage over bypassing the step.



A. Task 1

B. Task 2

C. Task 3

D. Task 4

Fig. 36. **Trajectories of A Successful Dropper Agent in Multiple Tasks.** A single dropper agent throwing markers evaporative markers (ρ =0.99) was successful in solving multiple tasks with various food locations. See Figure 25 for plotting conventions.

CHAPTER V

TASK III: TOPOLOGICAL NEUROEVOLUTION

A. Task Description

In the previous two experiments (Chapters III and IV), I have shown that external material interaction can represent spatial memory by creating an external recurrence. This suggests that it is viable that primitive animals without internal recurrence in the brain could have used such a marker dropping/detecting mechanism to meet their needs for memory. However, the results so far can only show what the primitive animals could have used in place of memory but not whether it was necessary. To answer this question, I need to prove the necessity of the use of environmental markers. Evolutionary advantage can mean the necessity of a mechanism in the evolution because what provides faster and higher adaptability will have higher chances for nature's selection. If external material interaction was indeed a necessary evolutionary step toward a fully internalized memory, it should have higher chances of being selected over direct evolution of recurrent connections bypassing this stage. Also, to have higher selection chance, it needs to have faster and higher adaptability.

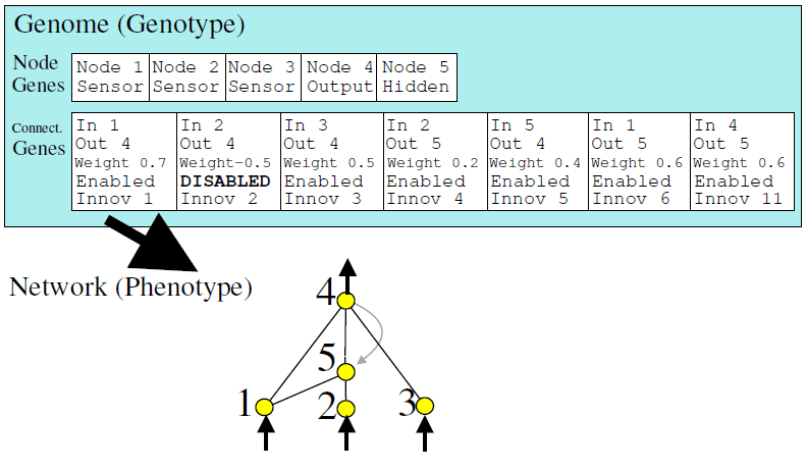
The experiment in the current chapter will focus on showing the evolutionary advantage of the dropper/detector mechanism over the direct memoryless (feedforward) to memory-equipped (recurrent) evolution. Because this experiment involves the emergence of recurrent connection from feedforward network, a topological evolution algorithm is required instead of traditional neuroevolution methods that evolves

only the weights on a fixed-topology. Before proceeding, I would like to present a little more detailed overview of the topological neuroevolution method I will use.

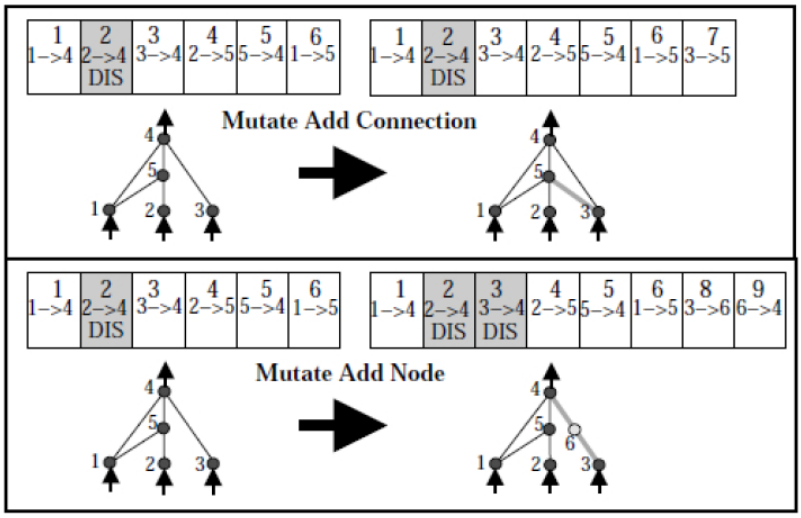
B. NEAT

Topological neuroevolution methods evolve both topology and weights of neural networks. Because natural evolution includes changes in the network topology in the brain, they mimic the natural evolution better than traditional weight-only neuroevolution methods. Moreover because the functionality of a neural network can be constrained by its topology, allowing the topology to evolve will set free the structural constraints and result in significant performance gain. Amongst many variation of such an approach, I chose NeuroEvolution of Augmenting Topologies (NEAT) because of its advantages over other topological evolution methods [51, 52, 53, 54].

Historical marking is the core of the NEAT algorithm. By enumerating each innovations, NEAT solves *competing conventions* problem, which is one of the main problems in neuroevolution [51, 54]. The crossover operation in NEAT happens between two genomes with identical *historical marking* or "innovation number", regardless of their locations and sizes. Moreover, NEAT keeps the size of resulting network from growing explosively by starting with the initial populations with the minimal structure. Figure 37 demonstrates the essence of the NEAT algorithm. It encodes the genome in two linear representations, node genes and connection genes (Figure 37-A). "Innovation number" is assigned to each connection genes according to the order of its appearance throughout the evolutionary mutation (Figure 37-B).



A. Genetic encoding



B. Mutation example

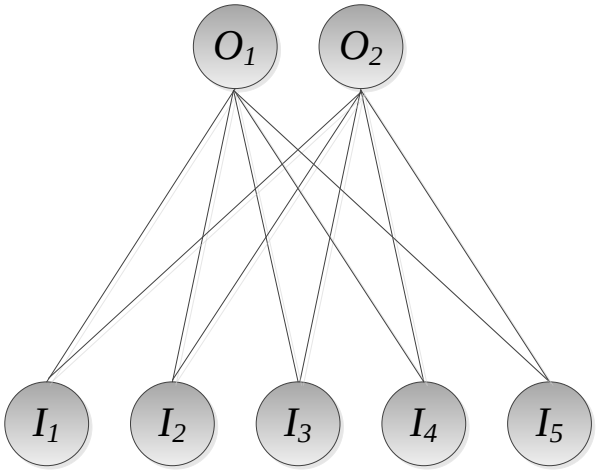
Fig. 37. **Neuroevolution of Augmenting Topologies (NEAT).** A. Genotype-phenotype matching in NEAT. NEAT uses "innovation numbers" to phenotypes in chromosomes. Regardless of their phenotypical representations, connections in two different individuals can be swapped during crossover. B. How "augmentation" happens is illustrated. Either connections (top) or nodes (bottom) can be added freely, hence the chromosome representation is variable length. Adapted from [54].

C. Methods

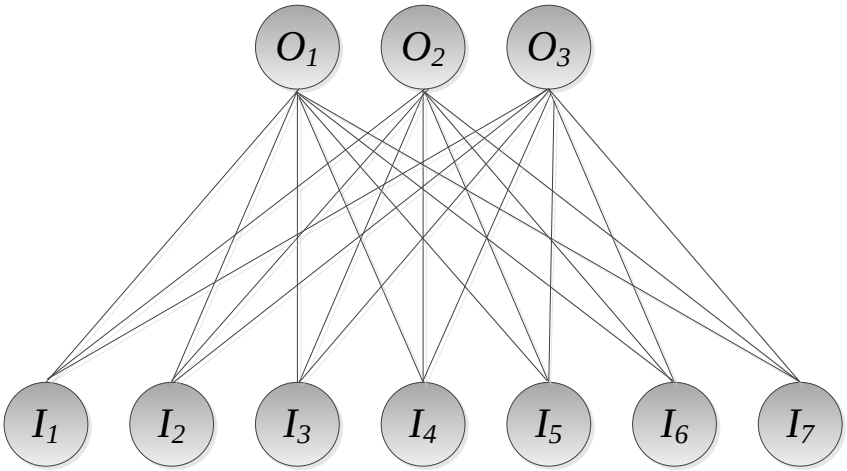
The memory task used in this experiment is identical to that in Chapter III. A small difference is that the agents are given only two pairs of falling balls (a total of 4 balls): one left-fast and one right-fast. Here, I compared the results of running NEAT algorithm on the initial neural networks with and without the dropper/detector mechanism. As mentioned earlier, the initial populations of NEAT need to have minimal structure. Figure 38 shows the initial topologies of the two types of networks. In the beginning, all populations have these identically minimal network topologies. As the evolution progresses, the population will diversify according to the NEAT topological mutation method (Figure 37-C).

D. Experiments and Results

30 evolutionary trials were performed on plain and dropper networks. As mentioned earlier, 2 pairs of balls with different falling speeds were given (pair 1: fast left and slow right, pair2: fast right and slow left). If an agent catches all 4 balls before the evolution reaches the preset maximum (500 generations), the trial is marked successful and the evolution is stopped. Fitness is identical to the previous ball-catching task in Chapter III. There are some NEAT-specific parameters worth mentioning. NEAT uses "speciation" to protect new topological innovations. Because topological mutation (add node or add connections) can cause a temporal decrease in the fitness score, it is unlikely for the newly evolved topology to be selected and survive to the next generation. Therefore, it is important to provide the new niches a time to opti-



A. Initial topology of plain neural network



B. Initial topology of dropper network

Fig. 38. **Initial Network Topologies for NEAT.** A. The initial topology of plain neural network only contains input and output layers of Figure 7. B. Initial dropper network for NEAT does not include the hidden layer. Similar to Figure 9, dropper/detector mechanism is implemented as additional inputs (I_6 and I_7 for detecting environmental markers) and an output (O_3 is for dropping environmental markers) units.

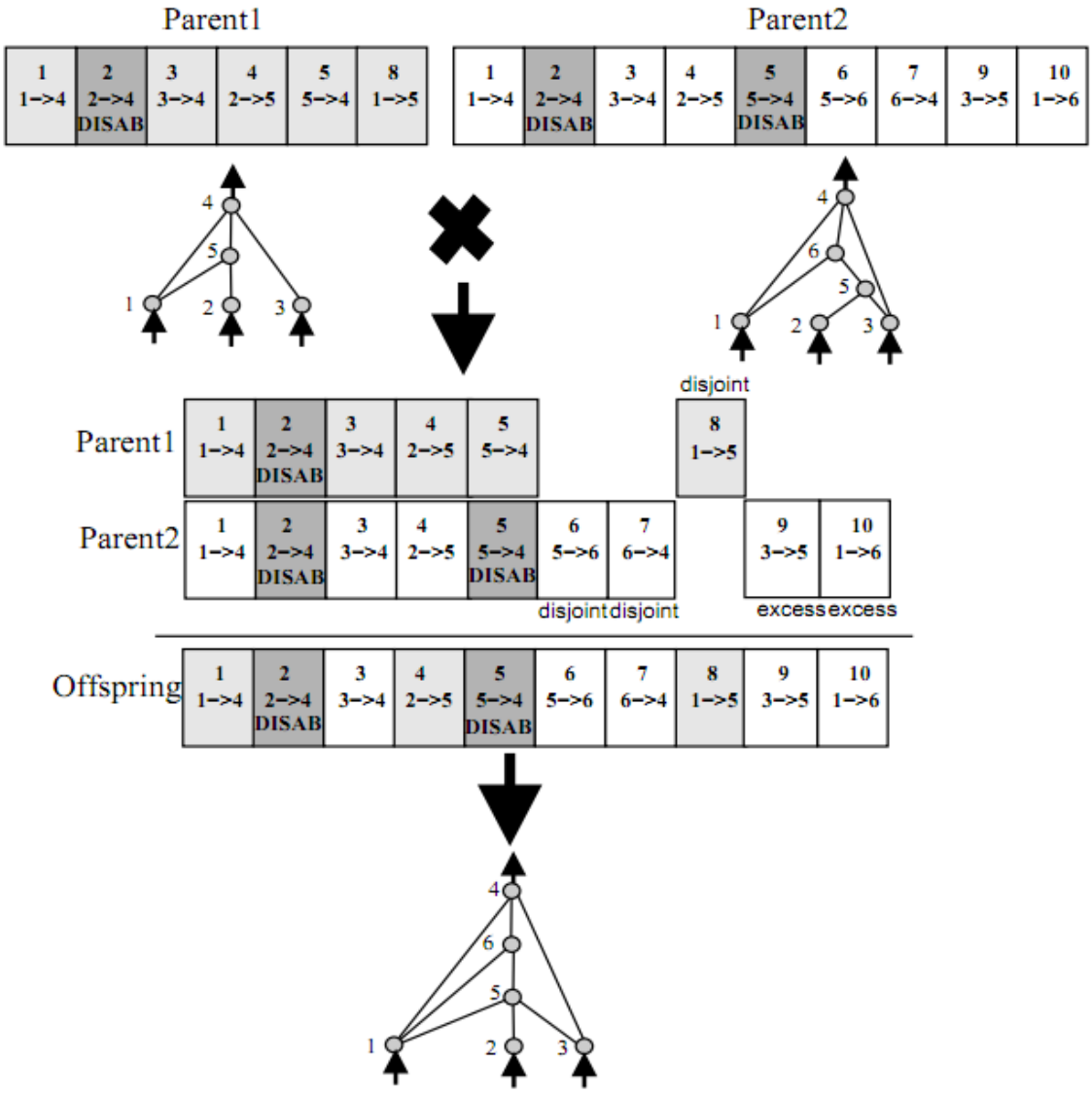


Fig. 39. **Gene Matching in NEAT.** Genomes with different topology can line up together according to their innovation numbers (shown at the top of each gene). *Disjoint* genes are those that do not match in the middle (genes of Parent 1 with innovation number 8 and of Parent 2 with innovation number 6 and 7), and *excess* genes are those that do not match at the end (genes of Parent 2 with innovation number 9 and 10). Remainings are *matching* genes (genes with innovation number 1-5 of both Parent 1 and 2). Adapted from [54].

mize, and NEAT solves this problem by allowing the competition for selection only among similar topologies. The equation below describes the measure of compatibility between two genomes:

$$\delta = c_1 E/N + c_2 D/N + c_3 \bar{W} \tag{5.1}$$

where δ is the compatibility measure, E is the number of excess genes, D is the number of disjoint genes, and $\bar{W}$ is the average weight differences of matching genes between the two genomes (see Figure 39). Table III lists the NEAT parameters used in my experiments.

Table III. Parameters for NEAT Algorithm

Parameter	Value
Maximum number of generations	500
Compatibility coefficient for excess genes(c_1 in 5.1)	1.0
Compatibility coefficient for disjoint genes(c_2 in 5.1)	1.0
Compatibility coefficient for weight differences of matching genes(c_3 in 5.1)	4.0
Compatibility threshold (δ_t)	3.0
General mutation rate	0.2
Add connection probability	0.07
Add recurrent connection probability	0.2
Add node probability	0.04
Crossover rate	0.7

Surprisingly, the result turned out that dropper network has an evolutionary potential much greater than the plain network without a dropping mechanism. While plain networks could not find a topology to solve the given task, dropper networks showed a fairly high success rate of 80.0% (24 out of 30). Moreover, dropper networks did not need much topological change to catch all 4 balls. As shown in the Table III, NEAT prefers to tune the weights of the current network to the topological mutation (probabilities of *add-node* and *add-connection* are only 0.04 and 0.07 respectively).

It turns out that the marker dropping/detecting mechanism is powerful enough to solve the ball-catching task with only a small number of topological mutation (or, sometimes with no topological mutation at all). Figure 40 shows the number of evolutionary steps required for the dropper networks to solve the task. On average, successful droppers required 38.67 topological evolutions. Figure 41 compares the convergence trends in the number of balls caught by both type of networks (plain and dropper) up to 300 evolutionary generations. Unlike the average number of catches in plain networks being stuck at around 2.75, the dropper networks show gradual increase in the number. The average numbers of active nodes, active connections, and active species of successful droppers were 10.7, 21.44, and 3.4 respectively. Considering that their initial values were 10, 21, and 1, successful droppers did not require much topological change to perform the given task.

Figure 42 shows the trajectories of top performances from each network type. The strategy used by typical successful dropper is shown in Figure 43. Usually, successful droppers have initial output O_3 (see Figure 38-B) greater than its randomly chosen marker threshold value. As a result, they begin to drop markers as soon as a ball is detected in either its left or right. The markers now denotes the previous movements. First, they tune network weights to respond to the falling ball. This way, they soon become able to catch 2 balls. Then, they tune network weights to adjust their response to the marker they throw.

The plain network evolved to catch only three balls whereas dropper network caught all four balls. The plain network uses the strategy of "default behavior".

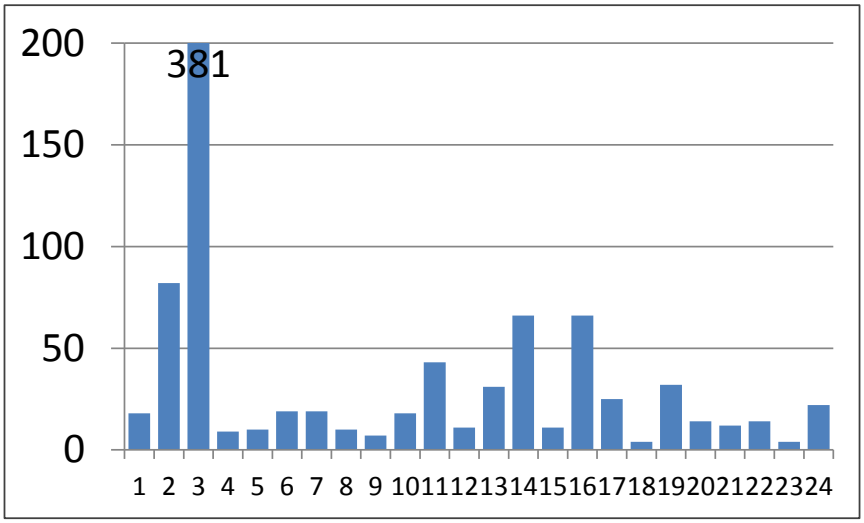


Fig. 40. **Number of Evolutions Required for Successful Dropper Networks.** Out of 30 trials, dropper networks solved the task 24 times (success rate 80.0%). The numbers of evolutionary steps for 24 successful droppers network are plotted. On the average, dropper networks required 38.67 steps of topological evolutions to find solutions. Compared to the topological evolution of plain networks which did not find a solution at all in 500 evolutionary steps, the dropper networks shows significantly higher success rate.

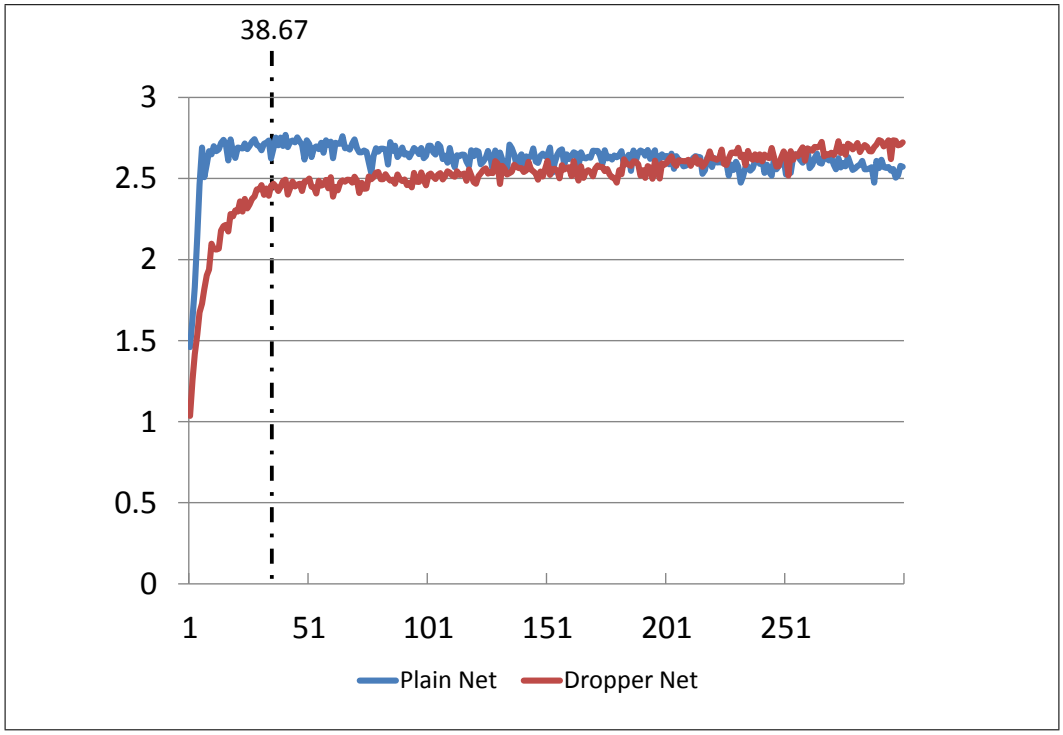


Fig. 41. **Performance Comparision in Topological Evolution.** X-axis denotes the evolutionary generations and y-axis denotes the average number of balls caught by the population. While the performance of plain network was stuck at about 2.75, the performance of dropper network keeps increasing. Dashed line denotes the average number of generations when dropper networks find solutions to catch all four balls (38.67).

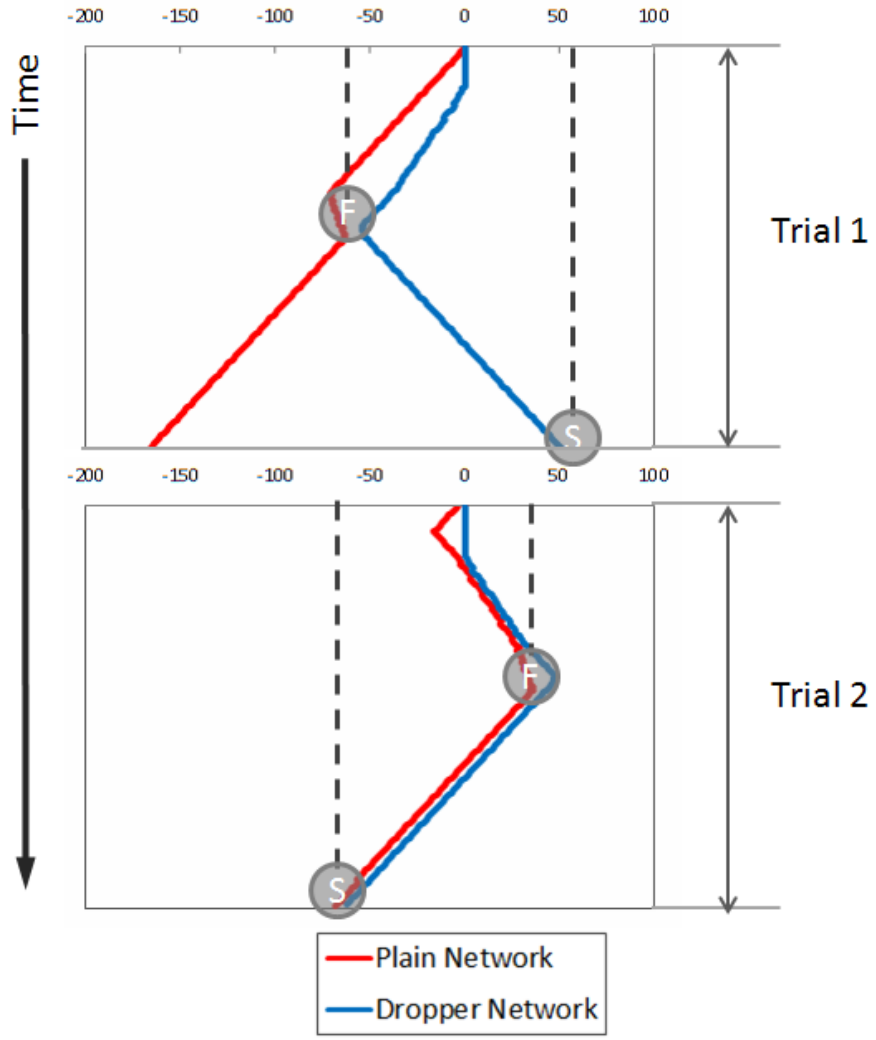


Fig. 42. **Agent Trajectories.** Two pairs of balls are fallen (fast-left & slow-right and fast-right & slow-left). Plain network successfully catches 3 balls and misses the slow-right ball in the first pair (red). Dropper network catches all 4 balls (blue).

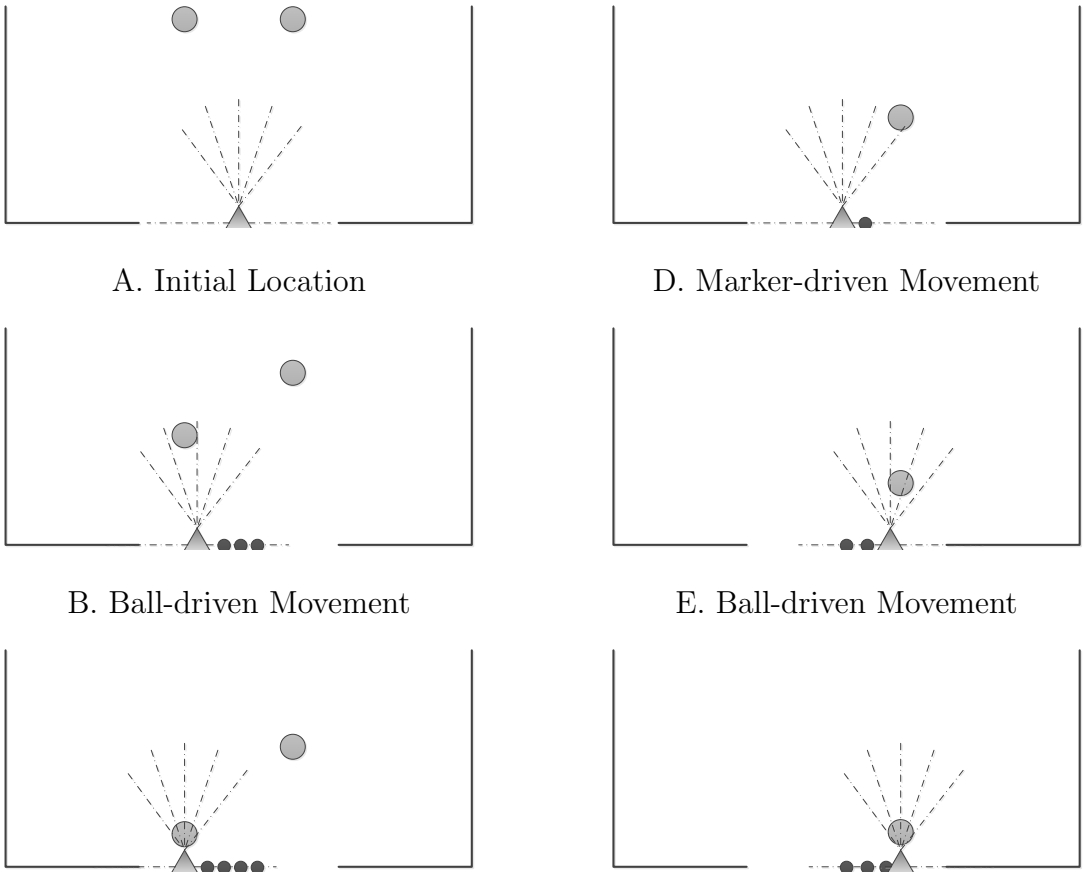


Fig. 43. **Dropper Strategy.** Detailed strategy for catching a pair of balls. A. Initial setting. B. Dropper agent starts to throw markers as it follows the fast-falling ball. C. Dropper agent catches fast-falling ball. D. As there is no input from ball, dropper agent follows markers until the slow-falling ball becomes detected again. E. Now that the slow-falling ball is detected, dropper agent generates *ball-driven* movement. Note that it begins to throw markers again. F. Dropper agent catches the slow-falling ball. This figure shows left-fast case only. Right-fast case is symmetrical.

When a ball is detected, it follows the ball. This way, it successfully catches the fast-falling balls in both pair. However when there is no input, it moves to one direction (in this case, left) rather than using the memory of the slow-falling balls. This strategy turns out to be successful at least half of the cases (fast-right & slow-left case). Recall that the reason the bias units are removed in the entire simulations is to stop such default behavior. It seems that this strategy using default behavior is very easy to learn (see the steep initial performance gain of plain networks in Figure 41), but it creates a "deep" local minimum, from which plain networks cannot escape with the current NEAT algorithm.

Another reason for the failure is the transience of internal recurrence. Whereas successful dropper networks have not evolved their topologies much (Figure 44), the resulting topologies of plain networks show much frequent internal recurrences created by NEAT (Figure 45). However, the number of recurrent connection is still very small compared to the recurrent network in Chapters III and IV. The fully-connected Elman tower contains $N_{hid}^2 \times N_{mem}$ recurrent connections, where N_{hid} is the number of hidden neurons and N_{mem} is memory order (tower height). Moreover, the extended input-to-output path exponentially weakens the effect of memory because distant inputs pass through more number of sigmoidal functions. Figure 46 examplifies the number of sigmoidal functions that signal of 8 step past in the first input node has to pass in a recurrent network created by NEAT (Figure 45). Note that in the Elman tower in Chapters III and IV, there is no sigmoidal function in the signal transition between memory units (from $H(t-1)$ to $H(t-N_{mem})$ in Figure 8). Indeed, the intuition

behind the choice of the Elman tower as a representative of recurrent network in previous experiments was to guarantee that the memory capacity is powerful enough for the given memory tasks.

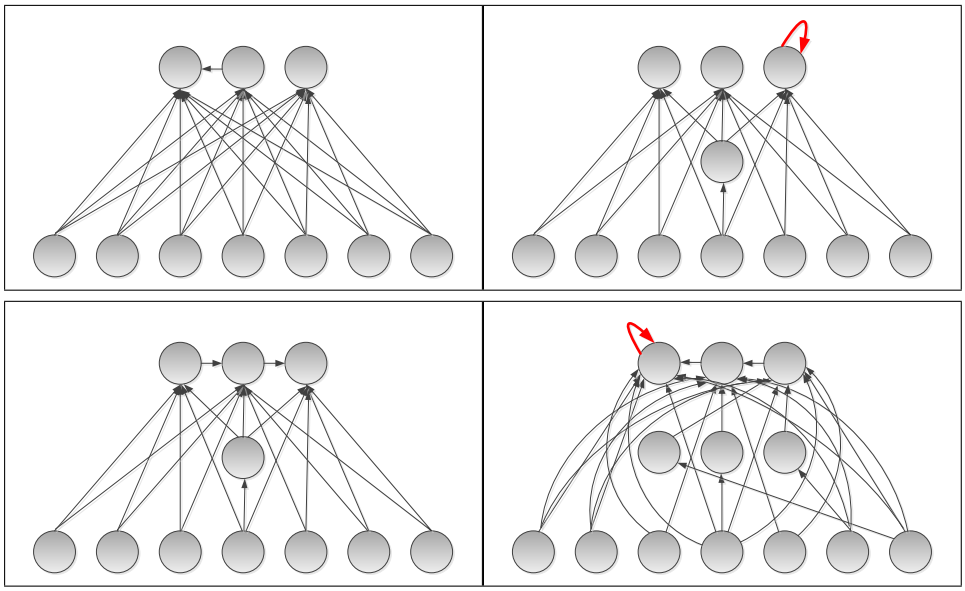


Fig. 44. **Topological Evolution of Dropper Networks.** A-D. Results of the topological evolution after solutions are found (500 generations) from the initial structure in Figure 38-B. Recurrent connections are highlighted in red.

E. Discussion and Conclusion

To summarize, the results of the experiment in this chapter showed that using environmental markers has benefit to the survival of the agents much higher than those without such mechanism. It showed that evolution of a complex internal memory capacity can be very difficult, but the marker dropping/detecting mechanism can make this evolutionary load very simple. Considering that the marker dropping and detecting capabilities are already given in primitive animals, it is likely that evolution

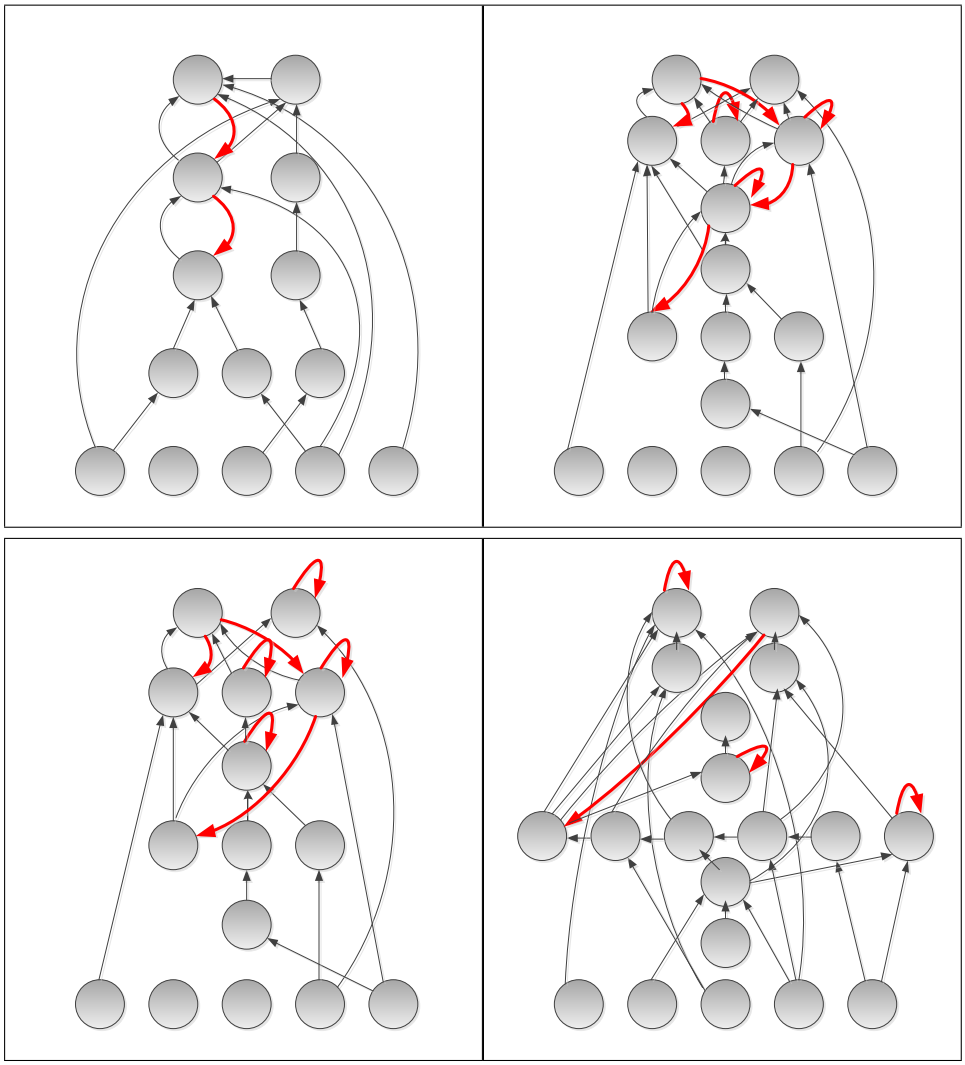


Fig. 45. **Topological Evolution of Plain Networks.** A-D. Results of the topological evolution after 500 generations from the initial structure in Figure 38-A. Recurrent connections are highlighted in red.

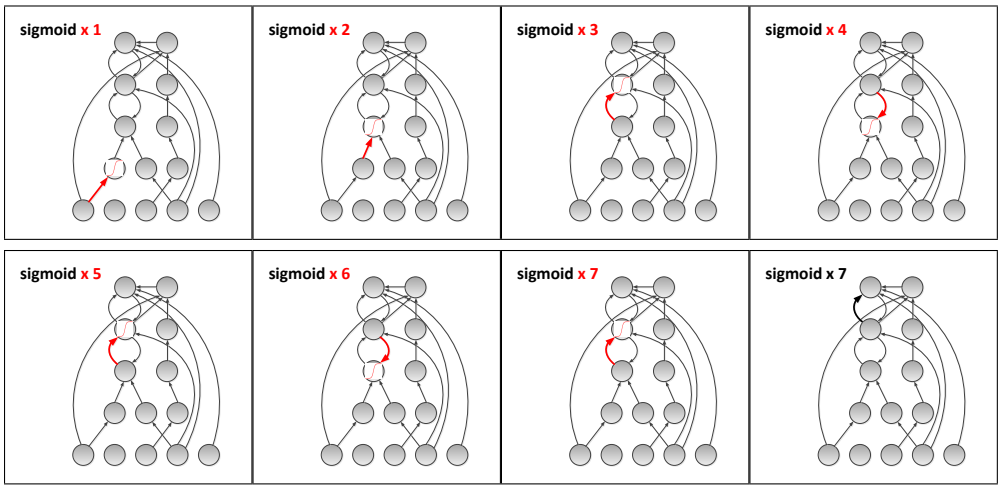


Fig. 46. **Number of Sigmoidal Functions in Input-to-output Pathway.** In one of the NEAT-generated recurrent network in Figure 45, inputs of 8 steps ahead needs to go by 7 sigmoidal functions. Signal transitions going by sigmoidal function is marked in red arrow. In Elman tower with memory order 7, inputs of 8 steps ahead only require two sigmoidal functions.

preferred marker interaction over direct internal recurrence in the process of memory development.

CHAPTER VI

DISCUSSION AND CONCLUSION

A. General Discussion

In search for the evolution of memory, I noticed the intricate relationship between memory and olfaction and hypothesized that self-generated olfactory cues could have served as external spatial memory. Using artificial evolution simulations, I have shown: (1) external material interaction generates performance comparable to that of internalized memory in a simple memory task (Chapter III); (2) external material interaction produces comparable or even more powerful performance when the volatile property of olfactory cue is called into play in a task critical for the survival of natural agent (Chapter IV); and (3) external marker interaction has clear evolutionary advantage and can be a must-taken step to an efficient internalized memory (Chapter V).

There are several existing works that share key mechanisms presented in my work, but there are important differences. Below, I will review three prominent approaches, (1) epistemic structures, (2) behavior-based robotics, and (3) ant colony optimization (and the use of artificial pheromones in general), and discuss how my work provides unique contributions.

The most notable is the work by Chandrasekharan and Stewart [61, 74] where "epistemic structures" (similar to pheromones) are dynamically deposited in the environment by agents, controlled by genetic algorithms or by Q-learning. In this case,

the goal was to reduce "tiredness" (cognitive load), so an explicit link to memory was not made. Furthermore, the task itself only contained one nest and one food source, thus memory capacity was not directly tested (see [75] for an extension of this to multiple food sources). Despite these differences, my work and that of Chandrasekharan and Stewart share the important concept that external markers can contribute to cognitive function even in reactive agents.

One tenet of behavior-based robotics is that instead of internalized representations, the environment itself can be used as a representation [7]. This allows the agents to adapt to environmental changes. I share this view, but go one step further. In behavior-based robotics, dynamic environment is usually viewed as an obstacle to solve, whereas in my work, the agent, although reactive in itself, purposefully alters the environment and utilizes the dynamics. It is only through this agent-environment coupling that the agent can break away from simple reactive behavior. Braitenberg and many other researchers have studied the importance of sensory-motor interaction between the agent and its environment in behavioral modification [76, 77, 49, 78]. It seems like it is only through this agent-environment coupling that the agent can break away from simple reactive behavior.

Another existing work with strong similarity to my work is the active field of ant colony optimization [59, 60, 62, 63, 64]. Ant colony optimization commonly uses artificial pheromone-like markers just like my dropper network does, and has been used successfully in many tough optimization tasks. However, the main use of pheromones in ant colony optimization is for social function (i.e., to enable communication among

different individuals in the swarm). This is in contrast with my own work presented in this paper where the pheromones (markers) are used for the agent's own individual purpose. Moreover, existing ant colony optimization works did not address a substantial property of the pheromone. Pheromones have two important properties: spatial and temporal. The location and time of a pheromone drop denote the locus and instant of importance, respectively. Nontheless, most studies on pheromone agents overlooked the temporal property by predetermining the pheromone deposition rules. Their agents throw pheromone unconditionally either at each state transition (online step-by-step pheromone trail update) or as they trace back their path after the solution is found (online delayed pheromone trail update) [79, 80]. If traditional pheromone agent with unconditional marker throwing behavior is used in the foraging task, the increased number of markers will create ambiguities. Similar to what the dropper networks with non-evaporative markers did to resolve ambiguity, they may also have to evolve more difficult strategies with additional rules. Therefore, it would be an interesting direction for future work to compare the performance between self-conditioned and unconditional marker throwing behaviors. Moreover, because the external markers with richer information can lower the complexity of the evolved strategy, it will also be interesting to confirm whether the strategy also gets simpler with additional types of markers.

B. Future Work

1. Linking with Neuroscience

As I briefly mentioned in the introduction, my work has strong implications on understanding the relationship between the olfactory system, the hippocampus, and the neuromodulator networks. In my view, olfaction is a form of external memory (dropper/detector network), while the hippocampus is a form of internal memory (recurrent network). Neuromodulators can be seen as an intermediate step between external and internal memory, where markers (neuromodulators) can be thought of as being dropped *inside* the brain, rather than outside, without explicit recurrent neuronal linkage (cf. [26]). Theoretical work also indicates that the boundary between the internal and the external can be blurred [81] (note: in this case, the main focus is on the reward structure).

2. Measuring Information Flow

Another line of future work is to gain insights on the exact computational nature of my dropper/detector agents by measuring the information flow in the sensorimotor loop, borrowing from [82]. Results in Figures 15, 16, 33, and 34 suggest that distributed memory can be reflected in the internal representation of dropper networks, which is similar to emergent meta-level cognition in dynamical systems [83, 65, 84]. My work shares a common view with these earlier works that embodiment plays an important role in forming such internally represented cognition. I will also scale up the memory capacity by introducing different types of markers and detectors, and to have multiple

agents interact with each other using these markers.

3. From External to Internal Memory

Figure 47 demonstrates the overall amplitude of the investigation of the evolutionary emergence of memory, which can be divided into the following three major steps:

1. Evolution of external memory use in reactive neural networks: Feed-forward networks only support reactive behavior. Memory-like behavior is expected through a minimal extension to this architecture to allow dropping and detecting environmental markers [48, 85, 86]. This type of agent-environment interaction is analogous to olfaction.

2. Internalized marker interaction: Once external markers are found to be effective in implementing memory, the next step is to test if such external marker interaction can be internalized, without introducing recurrent circuits. This could be analogous to the *neuromodulatory system* (see [26] for a review of the neuromodulatory system).

3. From internal marker interaction to recurrent circuits that support fully dynamic internal memory, such as in the *hippocampus* (cf. [87, 88]).

a. Evolution of Memory in Reactive Neural Networks with Dropper/Detector

The scope of the current dissertation encompasses the transition from the first to second panel of the big picture. This dissertation showed that interacting with self-produced external olfactory cues have higher evolutionary merit and it can be the

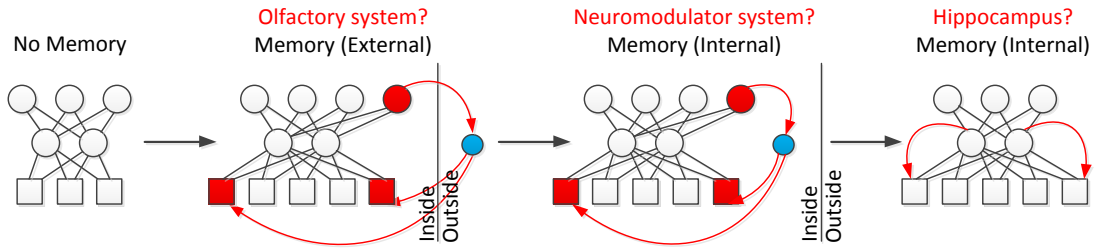


Fig. 47. **From the Present to the Past, and Forward to the Future.** Initially, only reactive behavior mediated by feedforward networks may have existed (left-most panel). (Note: "No memory" here means that the network is oblivious of its past input patterns.) By evolving external dropper/detector capability while maintaining the feedforward topology, simple memory function may have emerged (second panel), reminiscent of olfaction. Then, this kind of dropper/detector mechanism could have been internalized (third panel), leading to a full-blown recurrent architecture (fourth panel).

step for ancient animals to take to boost up their initial survival rate. To complete the quest to understand the full evolutionary path to the current, fully-internalized memory system, a series of research needs to be followed with a step-by-step progression.

b. Internalization of Marker Interaction

While external material interaction can be a good model for primitive animals, dependence on such dropping/detecting behavior may decrease as the animal becomes more advanced. The most advanced form would be where memory is implemented in recurrent circuits. However, a jump from a feedforward architecture to a heavily recurrent architecture could be very difficult. What could be a reasonable intermediate stage? My view is that the neuromodulatory system, with its broad diffuse targeting

and large variety of signaling molecules, can serve as a possible intermediate stage (for reviews, see [26, 89]), where neuromodulator secretion within the brain can serve as a surrogate of external marker dropping.

One possible necessity for this internalization can be found in the pheromonal communication process in Figure 4. Because olfactory cues are secreted in an open environment, they can potentially be detected by many different species as well as the conspecifics or the dropper themselves. When the eavesdroppers are heterospecifics, the pheromones are called *Kairomone*. In nature, there exist many cases of single-sided pheromone communication which only benefits the receiver (in this case, the evolution of pheromonal communication stops at "Spying" stage in Figure 4). Sometimes the eavesdroppers are benign or mutually beneficial to the pheromone droppers (in the latter case, the pheromone is also called *Synomone*), but sometimes predators can pry into them ([90, 25]). Then, the external material interaction becomes baleful to oneself, and could have driven its internalization.

In this step, an internal map that loosely models hippocampal place cells ([91, 92]) can serve as an internal, relative map of the external space, and mimic the neuromodulatory system to generate traces within the grid. ACh-like action will be used to direct attention on the internal grid as a feedforward neural network controller behaves in the 2D foraging task from Chapter IV. Recent findings suggest that this kind of attention is necessary for stable representation in the hippocampus ([93, 94] for an overview). Figure 48 shows an overview of the model. Note that the food sensors and marker sensors are not segregated rather than sharing the same space as

in Chapter IV.

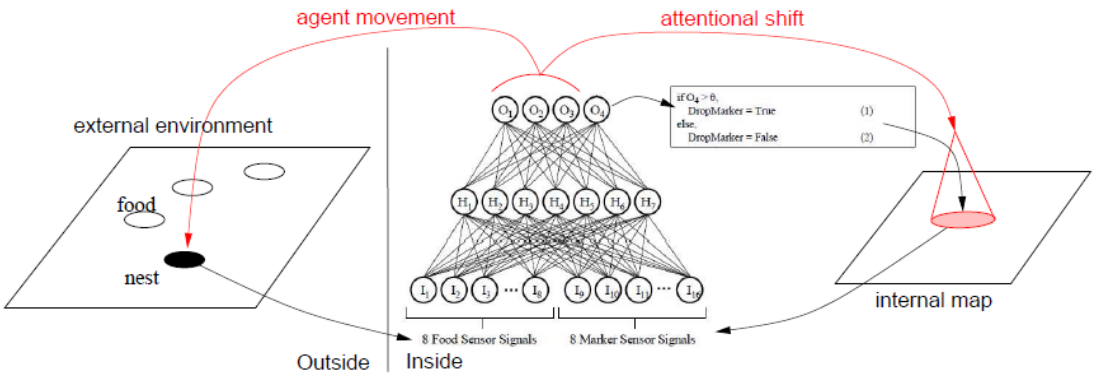


Fig. 48. **Neuromodulator-based Attention and Marking.** A feedforward network controller for the 2D foraging task is shown. It is the same as the one in Figure 22, except for an added internal map and attentional mechanism. Note that the internal map can be *significantly smaller* than the external environment it is modeling.

c. Evolution of Internal Recurrence

Once the pheromone dropping is internalized, generating internal recurrence from the internal dropper mechanism will be the last step to model the emergence of memory. Chapter V showed that plain NEAT cannot evolve the recurrent topology compatible to the dropper mechanism. Therefore, a significant extension is required on the original NEAT. One possibility is to introduce new genetic encoding elements besides nodes and connections. Then with new forms of mutation, the extended NEAT will switch between the loop created by internal marker dropping mechanism and direct recurrent connections. It is likely that a single recurrent connection will not suffice to substitute the neuromodulatory loop. Therefore, identification of the compatibility between the neuromodulatory loop and direct recurrent connection must be studied

as well.

It is also notable that recurrent neural architectures generally associated with memory of the past have the potential to evolve predictive internal dynamics projecting into the future needs [48, 95, 96].

C. Conclusion

In sum, the role and evolutionary benefit of the olfactory external material interaction has been studied in three artificial evolution simulations. This study can help us in understanding how the current animal brain came to be equipped with memory capability in a fully internalized form. Further research on the internalization of marker interaction and the establishment of internal recurrence can further elucidate a potential evolutionary route from memoryless to memory-equipped brain.

REFERENCES

[1] P. McCorduck, *Machines Who Think*. San Francisco, CA: W.H. Freeman and Company, 1979.

[2] D. Marr, "Artificial intelligence - a personal view," *Artificial Intelligence*, vol. 9, no. 1, pp. 37–48, 1977.

[3] R. A. Brooks and L. A. Stein, "Building brains for bodies," *Autonomous Robots*, vol. 1, no. 1, pp. 7–25, 1994.

[4] N. J. Nilsson, "Human-level artificial intelligence? Be serious!" *AI Magazine*, vol. 26, no. 4, pp. 68–75, 2006.

[5] J. McCarthy, "What is artificial intelligence?" November 2007, computer Science Department, Stanford University, Stanford, CA. [Online]. Available: http://www-formal.stanford.edu/jmc/whatisai/

[6] G. Marcus, *Kluge*. Boston, MA: Houghton Mifflin, 2008.

[7] R. A. Brooks, "Intelligence without representation," *Artificial Intel*, vol. 47, no. 1-3, pp. 139–159, 1991.

[8] R. D. Beer and J. C. Gallagher, "Evolving dynamical neural networks for adaptive behavior," *Adaptive Behavior*, vol. 1, no. 1, pp. 91–122, 1992.

[9] P. Husbands, I. Harvey, D. Cliff, and G. Miller, "Artificial evolution: A new path for artificial intelligence?" *Brain and Cognition*, vol. 34, no. 1, pp. 130–159,

1997.

[10] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence through Simulated Evolution*. New York: John Wiley, 1966.

[11] D. B. Fogel, *Evolutionary Computation*, 2nd ed. New York: IEEE Press, 1999.

[12] E. B. Bolton, "The relation of memory to intelligence," *Journal of Experimental Psychology*, vol. 14, no. 1, pp. 37–67, 1931.

[13] J. Piaget and B. Inhelder, *Memory and Intelligence*. New York: Basic Books, 1973.

[14] M. I. Jordan, *Attractor Dynamics and Parallelism in a Connectionist Sequential Machine*. Piscataway, NJ: IEEE Press, 1990.

[15] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[16] D. J. Felleman and D. C. Van Essen, "Distributed hierarchical processing in primate cerebral cortex," *Cerebral Cortex*, vol. 1, no. 1, pp. 1–47, 1991.

[17] K. G. Field, G. J. Olsen, D. J. Lane, S. J. Giovannoni, M. T. Ghiselin, E. C. Raff, N. R. Pace, and R. A. Raff, "Molecular phylogeny of the animal kingdom," *Science*, vol. 239, no. 4841, pp. 748–753, 1988.

[18] N. King, "The unicellular ancestry of animal development," *Developmental Cell*, vol. 7, no. 3, pp. 313–325, 2004.

[19] J. G. Hildebrand, "Analysis of chemical signals by nervous systems," in *Proceedings of National Academy of Sciences*, vol. 92, 1995, pp. 67–74.

[20] G. Smith, *Primitive Animals*. London, UK: Cambridge University Press, 1911.

[21] G. H. Wadhams and J. P. Armitage, "Making sense of it all: Bacterial chemotaxis," *Nature Reviews Molecular Cell Biology*, vol. 5, no. 12, pp. 1024–1037, 2004.

[22] Z. Tang-Martinez, "The mechanisms of kin discrimination and the evolution of kin recognition in vertebrates: A critical re-evaluation," *Behavioural Processes*, vol. 53, no. 1-2, pp. 21–40, 2001.

[23] D. W. Pfennig and P. W. Sherman, "Kin recognition," *Scientific American*, vol. 272, no. 6, pp. 98–103, 1995.

[24] P. W. Sorensen and N. E. Stacey, "Evolution and specialization of fish hormonal pheromones," in *Advances in Chemical Signals in Vertebrates*, R. E. Johnston, D. Müller-Schwarze, and P. W. Sorensen, Eds. New York: Plenum Publishers, 1999, pp. 15–47.

[25] T. D. Wyatt, *Pheromones and Animal Behavior*. Cambridge, U.K.: Cambridge University Press, 2003.

[26] J. L. Krichmar, "The neuromodulatory system: A framework for survival and adaptive behavior in a challenging world," *Adaptive Behavior*, vol. 16, no. 6, pp. 385–399, 2008.

[27] R. S. Herz and T. Engen, "Odor memory: Review and analysis," *Psychonomic Bulletin & Review*, vol. 3, no. 3, pp. 300–313, 1996.

[28] M. S. Gordon and E. C. Olson, *Invasions of the Land: The Transitions of Organisms from Aquatic to Terrestrial Life*. New York: Columbia University Press, 1995.

[29] Y. Niimura and M. Nei, "Evolutionary dynamics of olfactory receptor genes in fishes and tetrapods," in *Proceedings of the National Academy of Sciences*, vol. 102, no. 17, 2005, pp. 6039–6044.

[30] ——, "Extensive gains and losses of olfactory receptor genes in mammalian evolution," *Public Library of Science ONE*, vol. 2, no. 8, p. e708, 2007.

[31] D. Müller-Schwarze, "Signal specialization and evolution in mammals," in *Advances in Chemical Signals in Vertebrates*, R. E. Johnston, D. Müller-Schwarze, and P. W. Sorensen, Eds. New York: Plenum Publishers, 1999, pp. 1–14.

[32] D. G. Moulton, "Olfaction in mammals," *American Zoologist*, vol. 7, no. 3, pp. 421–429, 1967.

[33] Y. Gilad, V. Wiebe, M. Przeworski, D. Lancet, and S. Pääbo, "Loss of olfactory receptor genes coincides with the acquisition of full trichromatic vision in primates," *Public Library of Science Biology*, vol. 2, no. 1, pp. 0120–0125, 2004.

[34] M. Doop, C. Mohr, B. Folley, W. Brewer, and S. Park, "Olfaction and memory," in *Olfaction and the Brain*, W. J. Brewer, D. Castle, and C. Pantelis, Eds.

Cambridge, U.K.: Cambridge University Press, 2006, pp. 65–82.

[35] A. Pontius, "Psychiatry - explaining the unexplainable," April 1996, news from Harvard Medical, Dental and Public Health Schools. [Online]. Available: http://archives.focus.hms.harvard.edu/1996/Apr26_1996/Psychiatry.html

[36] D. C. Rubin, E. Groth, and D. J. Goldsmith, "Olfactory cuing of autobiographical memory," *The American Journal of Psychology*, vol. 97, no. 4, pp. 493–507, 1984.

[37] R. S. Herz and G. C. Cupchik, "An experimental characterization of odor-evoked memories in humans," *Chemical Senses*, vol. 17, no. 5, pp. 519–528, 1992.

[38] S. Chu and J. J. Downes, "Proust nose best: Odors are better cues of autobiographical memory," *Memory & Cognition*, vol. 30, no. 4, pp. 511–518, 2002.

[39] A.-M. Mouly, U. Kindermann, R. Gervais, and A. Holley, "Involvement of the olfactory bulb in consolidation processes associated with long-term memory in rats," *Behavioral Neuroscience*, vol. 107, no. 3, pp. 451–457, 1993.

[40] F. R. Schab, "Odor memory: Taking stock," *Psychology Bulletin*, vol. 109, no. 2, pp. 242–251, 1991.

[41] G. M. Zucco, "Anomalies in cognition: Olfactory memory," *European Psychologist*, vol. 8, no. 2, pp. 77–86, 2002.

[42] J. Altman, "Autoradiographic and histological studies of postnatal neurogenesis. IV. Cell proliferation and migration in the anterior forebrain, with special

reference to persisting neurogenesis in the olfactory bulb," *The Journal of Comparative Neurology*, vol. 137, no. IV, pp. 433–457, 1969.

[43] J. Frisén, C. B. Johansson, C. Lothian, and U. Lendahl, "Central nervous system stem cells in the embryo and adult," *CMLS, Cellular and Molecular Life Science*, vol. 54, no. 9, pp. 935–945, 1998.

[44] R. Machold, S. Hayashi, M. Rutlin, M. D. Muzumdar, S. Nery, J. G. Corbin, A. Gritli-Linde, T. Dellovade, J. A. Porter, S. L. Rubin, H. Dudek, A. P. McMahon, and G. Fishell, "Sonic hedgehog is required for progenitor cell maintenance in telencephalic stem cell niches," *Neuron*, vol. 39, no. 6, pp. 937–950, 2003.

[45] V. Palma, D. A. Lim, N. Dahmane, P. Sánchez, T. C. Brionne, C. D. Herzberg, Y. Gitton, A. Carleton, A. Álvarez Buylla, and A. R. Altaba, "Sonic hedgehog controls stem cell behavior in the postnatal and adult brain," *Development*, vol. 132, pp. 335–344, 2004.

[46] M. J. Matarić, "Behavior-based robotics as a tool for synthesis of artificial behavior and analysis of natural behavior," *Trends in Cognitive Science*, vol. 2, no. 3, pp. 82–87, 1998.

[47] R. Beckers, O. E. Holland, and J. L. Deneubourg, "From local actions to global tasks: Stigmergy and collective robotics," in *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, R. Brooks and P. Maes, Eds. The MIT Press, 1994, pp. 181–189.

[48] J. R. Chung, J. Kwon, and Y. Choe, "Evolution of recollection and prediction in neural networks," in *Proceedings of the International Joint Conference on Neural Networks*. Piscataway, NJ: IEEE Press, 2009, pp. 571–577.

[49] S. Nolfi, "Evolutionary robotics: Exploiting the full power of self-organization," *Connection Science*, vol. 10, no. 3-4, pp. 167–184, 1998.

[50] D. Floreano, "Reducing human design and increasing adaptability in evolutionary robotics," in *Evolutionary Robotics: From Intelligent Robots to Artificial Life*, T. Gomi, Ed. Ontario, Canada: AAI Books, 1997.

[51] D. Whitley, "Genetic reinforcement learning for neurocontrol problems," *Machine Learning*, vol. 13, no. 2-3, pp. 259–284, 1993.

[52] F. Gruau, D. Whitley, and L. Pyeatt, "A comparison between cellular encoding and direct encoding for genetic neural networks," in *Genetic Programming 1996: Proceedings of the First Annual Conference*, J. R. Koza, D. E. Goldberg, D. B. Fogel, and R. L. Riolo, Eds. Cambridge, MA: The MIT Press, 1996, pp. 81–89.

[53] X. Yao, "Evolving artificial neural networks," in *Proceedings of the IEEE*, vol. 87, no. 9, 1999, pp. 1423–1447.

[54] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.

[55] F. Gomez and R. Miikkulainen, "Incremental evolution of complex general behavior," *Adaptive Behavior*, vol. 5, no. 3-4, pp. 317–342, 1997.

[56] W.-P. Lee, J. Hallam, and H. H. Lund, "Learning complex robot behaviours by evolutionary computing with task decomposition," *Lecture Notes in Computer Science*, vol. 1545, pp. 155–172, 1998.

[57] S. W. Stepniewski and A. J. Keane, "Topology design of feedforward neural networks by genetic algorithms," *Lecture Notes in Computer Science*, vol. 1141, pp. 771–780, 1996.

[58] T. D. Jorgensen, B. P. Haynes, and C. C. F. Norlund, "Evolving artificial neural networks through topological complexification," *Engineering Letters*, vol. 17, no. 1, pp. 21–29, 2008.

[59] I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein, "Distributed covering by ant-robots using evaporating traces," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 5, pp. 918–933, 1999.

[60] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behaviour," *Nature*, vol. 406, pp. 39–42, 2000.

[61] S. Chandrasekharan and T. Stewart, "Reactive agents learn to add epistemic structures to the world," in *CogSci2004*, K. D. Forbus, D. Gentner, and T. Regier, Eds. Hillsdale, NJ: Lawrence Erlbaum, 2004.

[62] L. Panait and S. Luke, "A pheromone-based utility model for collaborative foraging," in *Proceedings of the Third International Joint Conference on Autonomous*

Agents and Multiagent Systems (AAMAS '04). Washington DC: IEEE Computer Society, 2004, pp. 36–43.

[63] T. Ziemke, N. Bergfeldt, G. Buason, T. Susi, and H. Svensson, "Evolving cognitive scaffolding and environment adaptation: A new research direction for evolutionary robotics," *Connection Science*, vol. 16, no. 4, pp. 339–350, 2004.

[64] M. Dorigo and C. Blum, "Ant colony optimization theory: A survey," *Theoretical Computer Science*, vol. 344, no. 2-3, pp. 243–278, 2005.

[65] R. D. Beer, "The dynamics of active categorical perception in an evolved model agent," *Adaptive Behavior*, vol. 11, no. 4, pp. 209–243, 2003.

[66] R. Ward and R. Ward, "2006 special issue: Cognitive conflict without explicit conflict monitoring in a dynamical agent," *Neural Networks*, vol. 19, no. 9, pp. 1430–1436, 2006.

[67] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*. Boulder, CO: Westview Press, 1991.

[68] P. A. Hetherington and M. L. Shapiro, "A simple network model simulates hippocampal place fields: II. Computing goal-directed trajectories and memory fields," *Behavioral Neuroscience*, vol. 107, no. 3, pp. 434–443, 1993.

[69] L. M. Rocha, "Eigenbehavior and symbols," *Systems Research*, vol. 12, no. 3, pp. 371–384, 1996.

[70] A. F. Skutch, "Do tropical birds rear as many young as they can nourish?" *Ibis*, vol. 91, pp. 430–455, 1949.

[71] C. R. Carroll and D. H. Janzen, "Ecology of foraging by ants," *Annual Review of Ecology and Systematics*, vol. 4, pp. 231–257, 1973.

[72] C. C. Smith and O. J. Reichman, "The evolution of food caching by birds and mammals," *Annual Review of Ecology and Systematics*, vol. 15, pp. 329–351, 1984.

[73] A. D. Baddeley and G. J. Hitch, "The recency effect: Implicit learning with explicit retrieval?" *Memory and Cognition*, vol. 21, no. 2, pp. 146–155, 1993.

[74] S. Chandrasekharan and T. C. Stewart, "The origin of epistemic structures and proto-representations," *Adaptive Behavior*, vol. 15, no. 3, pp. 329–353, 2007.

[75] A. E. A. Goosen, "The world inside your head: From structuring torepresentations to language," Master's thesis, Department of Artificial Intelligence, Radboud University Nijmegen, 2008.

[76] V. Braitenberg, *Vehicles: Experiments in Synthetic Psychology.* Cambridge, MA: MIT Press, 1984.

[77] R. Pfeifer and C. Scheier, "Sensory-motor coordination: The metaphor and beyond," *Robotics and Autonomous Systems*, vol. 20, no. 2-4, pp. 157–178, 1997.

[78] C. H. Yong and R. Miikkulainen, "Coevolution of role-based cooperation in multiagent systems," *IEEE Transactions on Autonomous Mental Development*, vol. 1,

pp. 170–186, 2009.

[79] O. Cordón, F. Herrera, and T. Stützle, "A review on the ant colony optimization metaheuristic: Basis, models and new trends," *Mathware and Soft Computing*, vol. 9, no. 2-3, pp. 141–175, 2002.

[80] R. J. Mullen, D. Monekosso, S. Barman, and P. Remagnino, "A review of ant algorithms," *Expert Systems with Applications*, vol. 36, no. 6, pp. 9608–9617, 2009.

[81] S. Singh, R. L. Lewis, A. G. Barto, and J. Sorg, "Intrinsically motivated reinforcement learning: An evolutionary perspective," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 2, pp. 70–82, 2010.

[82] M. Lungarella and O. Sporns, "Mapping information flow in sensorimotor networks," *Public Library of Science Biology*, vol. 2, no. 10, pp. 1301–1312, 2006.

[83] R. D. Beer, "Dynamical approaches to cognitive science," *Trends in Cognitive Sciences*, vol. 4, no. 3, pp. 91–99, 2000.

[84] M. Maniadakis and J. Tani, "Dynamical systems account form meta-level cognition," in *SAB '08 Proceedings of the 10th International Conference on the Simuation of Adaptive Behavior: From Animals to Animats.* Heidelberg, Germany: Springer-Verlag Berlin, 2008, pp. 311–320.

[85] J. R. Chung and Y. Choe, "Emergence of memory-like behavior in reactive agents using external markers," in *21st International Conference on Tools with Artificial*

Intelligence (ICTAI09), Newark, NJ, USA, November 2-5 2009.

[86] J. R. Chung, J. Kwon, T. A. Mann, and Y. Choe, "Evolution of time in neural networks: From the present to the past, and forward to the future," in *The relevance of the time domain to neural network models*, A. R. Rao and G. A. Cecchi, Eds. Berlin, Germany: Springer, 2011, in Press.

[87] D. G. Amaral and M. P. Witter, "The three-dimensional organization of the hippocampal formation: A review of anatomical data," *Neuroscience*, vol. 31, no. 3, pp. 571–591, 1989.

[88] G. V. Wallenstein, M. E. Hasselmo, and H. Eichenbaum, "The hippocampus as an associator of discontiguous events," *Trends in Neurosciences*, vol. 21, no. 8, pp. 317–323, 1998.

[89] O. Sporns and W. H. Alexander, "Neuromodulation and plasticity in an autonomous robot," *Neural Networks*, vol. 15, no. 4-6, pp. 761–774, 2002.

[90] D. L. Wood, "The role of pheromones, kairomones, and allomones in the host selection and colonization behavior of bark beetles," *Annual Review of Entomology*, vol. 27, pp. 411–446, 1982.

[91] J. O'Keefe and D. H. Conway, "Hippocampal place units in the freely moving rat: Why they fire where they fire," *Experimental Brain Research*, vol. 31, no. 4, pp. 573–590, 1978.

[92] J. O'Keefe, N. Burgess, J. G. Donnett, K. J. Jeffery, and E. A. Maguire, "Place

cells, navigational accuracy, and the human hippocampus," *Philosophical Transactions of the Royal Society of London (B): Biological Sciences*, vol. 353, no. B, pp. 1333–1340, 1998.

[93] C. G. Kentros, N. T. Agnihotri, S. Streater, R. D. Hawkins, and E. R. Kandel, "Increased attention to spatial context increases both place field stability and spatial memory," *Neuron*, vol. 42, no. 2, pp. 283–295, 2004.

[94] E. I. Moser, "Hippocampal place cells demand attention," *Neuron*, vol. 42, no. 2, pp. 183–185, 2004.

[95] J. Kwon and Y. Choe, "Internal state predictability as an evolutionary precursor of self-awareness and agency," in *Proceedings of the Seventh International Conference on Developoment and Learning*. IEEE, 2008, pp. 109–114.

[96] ——, "Predictive internal neural dynamics for delay compensation," in *Second World Congress on Nature and Biologically Inspired Computing (NaBIC2010)*, 2010, pp. 443–448, in press.

VITA

Ji Ryang Chung received his Bachelor of Science degree in computer science and engineering from Seoul National University in 2004. He is expected to receive his Doctor of Philosophy degree in computer science and engineering at Texas A&M University, College Station, in May 2012. His research interests include the study of the evolutionary route of internal memory systems and the application of the concept to autonomous agents and multiagent systems.

Permanent Address: 416, Maetan 3-dong

Yeongtong-gu, Suwon-si

Gyeonggi-do 443-742

Republic of Korea